Volume 48 Number 4 December 2024 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Middle-European Conference on Applied Theoretical Computer Science (MATCOS-22)

Guest Editors:

Andrej Brodnik,
Gábor Galambos,
Attila Sali

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Guest Editorial Preface

Special issue on Middle-European Conference on Applied Theoretical Computer Science - MATCOS-22

We are happy to introduce you to a special section on the "mini" conference MATCOS-22 that was held in Koper, Slovenia as part of the multi-conference Information Society on October 13th and 14th 2022. After the 2019 edition that was marked by COVID situation, in 2022 we again got together in person, which made the event much more fruitful. The conference is a result of a long-lasting collaboration between University of Primorska and University of Szeged and also one of the activities in a joint Slovenian-Hungarian research project N2-0171 *Graph theory and combinatorial scientific computing*.

The conference was started by the invited talk "*Interpretability of deep-learned error-correcting codes*" delivered by György Turán from the University of Illinois at Chicago and Research Group on AI, University of Szeged, which was followed by 10 sessions on topics of Machine learning, Algorithm's complexity, Graph Theory, Optimization, Discrete Optimization, and Parallel computation. For the conference 32 papers were submitted, out of which after the reviewing process the programme committee choose 12 regular papers for presentation. The remaining papers were presented as short contributions.─Authors from 7 countries were participating at the conference.

At the end of conference, the speakers were offered to prepare a full papr to be published as a special section of Informatica. The submitted papers went again through a thorough reviewing proces which resulted in six papers that are in front of you. They are addressing three different areas presented at the conference. First, Alain Quilliot (*Surrogate Estimators for Collaborative Decision*), and Hasmik Sahakyan, Gyula Katona, and Levon Aslanyan (*Study on Using Reinforcement Learning for the Monotone Boolean Reconstruction*) touch upon the topic of the machine learning. Next, Andrej Brodnik, Bengt J. Nilsson, and Gordana Vujović (*Online Bin Covering with Exact Parameter Advice*) and Eszter Csókás, and Tamás Vinkó (*A heuristic for influence maximization under deterministic linear threshold model*) are addressing optimisation topics. Finally, Tomaž Dobravec (*Empirical Evaluation of Algorithm Performance: Addressing Execution Time Measurement Challenges*) and Attila Sali and Munqath Alatar (*Approximate Keys and Functional Dependencies in Incomplete Databases with Limited Domains–Algorithmic Perspective*) study the complexity of algorithms.

At last, but not least we want to thank Balázs Dávid, Branko Kavšek, Matjaž Krnc, and Rok Požar for all their contribution as a local organizing committee.

We hope you will enjoy the section on MATCOS-22 and looking forward to see you on MATCOS 2025!

Koper, December 2024

Guest Editors

Andrej Brodnik

(andrej.brodnik@upr.si), Faculty of Computer and Information Science, University of Ljubljana

Gábor Galambos

(galambos@jgypk.szte.hu), Department of Informatics Applications, Institute of Applied Natural Sciences

Attila Sali

(saliattila@gmail.com), Alfréd Rényi Institute of Mathematics Hungarian Academy of Sciences

https://doi.org/10.31449/inf.v48i4.4655 Informatica **48** (2024) 507–512 **507**

Algorithmic Perspective of Strongly Possible Keys and Functional Dependencies

Munqath Alattar[1], Attila Sali[2,3]
[1]ITRDC, University of Kufa, Iraq
[2]Alfréd Rényi Institute of Mathematics, Budapest, Hungary
[3]Department of Computer Science, Budapest University of Technology and Economics, Budapest, Hungary
E-mail: munqith.alattar@uokufa.edu.iq, sali.attila@renyi.hu

Keywords: Functional dependencies, database key, missing values, approximate dependencies, approximate keys

Received: February 3, 2023

It is common to encounter missing values in database tables. For an incomplete table, a possible world can be obtained by replacing any missed value with a value from the attribute (infinite) domain. A possible key (possible functional dependency) is satisfied in an incomplete table "T" if there exists a possible world of "T" that satisfies the key (the functional dependency) constraint. If all possible worlds of "T" satisfy the key (functional dependency), then we say that "T" satisfies a certain key (functional dependency). The concept of strongly possible worlds was introduced recently that considers only the active domain (the set of values that are already appearing in each attribute in the table), in a way that a strongly possible world is obtained by replacing any missing value with a value from the corresponding attributes active domain. So, a strongly possible key spKey (functional dependency spFD) is satisfied by a table "T" if there exists a strongly possible world that satisfies the key (functional dependency). In this paper, we investigate the approximation measures of spKeys and spFDs when the strongly possible constraint is not satisfied by a given table. We introduce the g_5 measures which is the ratio of the minimum number of tuples that need to be added so that the constraint is satisfied. The measure g_3 represent the ratio of the minimum number of tuples that need to be removed so that the table satisfies the constraint. We introduce a new measure g_5, which is the ratio of the minimum number of tuples to be added to the table so the result satisfies the constraint. Where adding new tuples with new values will extend the active domain. We prove that g_3 is an upper bound of g_5 for a constraint in a table. Furthermore, g_3 and g_5 are independent of each other, where there exist tables of some large number of tuples that satisfy $g_3 - g_5 = \frac{p}{q}$ for any rational number $0 \leq \frac{p}{q} < 1$. We study the complexity of determining these approximate measures.

Povzetek: Raziskane so aproksimacijske mere za močne možne ključe (spKeys) in funkcionalne odvisnosti (spFDs). Predstavljen je nov kazalnik g5, ki ocenjuje dodajanje n-tork za izpolnitev omejitev, s poudarkom na računski zapletenosti in uporabnosti v podatkovnih bazah.

1 Introduction

There are many reasons that may cause the missingness of values in the industrial and research databases, such as data maintenance, errors during data entry, surveys, and so on [8]. Imputation and deletion are the main two approaches to handle the missing values problem in a database. Imputation (assignment of a value to the occurrence of any NULL) is the main approach to handle missing value problem [13]. In [3], an imputation method was introduced that replaces the missing value using only the shown information in the table (which is defined as the active domain of that attribute), and we call the complete table achieved by this method a *strongly possible world*. The reason we consider only the values that are shown in a table for imputation is that it is not always proper to consider values that are not shown in the table.

Using this concept, strongly possible keys (spKeys) and strongly possible functional dependencies (spFDs) were defined in [5, 4] as new key and functional dependency constraints that are satisfied by replacing missing values (NULL) with values that are shown in the corresponding attribute. The formal definitions of spKeys and spFDs are provided in Section 2. In this paper, we continue the work started in [5] which introduced an approximation measure that calculates how close a set of attributes in a table can form a key if they are not. An active domain may not contain enough values to replace NULLs to make all resulting tuples distinct from each other on a key K, so, removing some tuples can be a solution. This paper studies an approximation measures of spKeys and spFDs by adding tuples not removing them, where adding a tuple with new unique values that are not shown before adds more values to the attributes' active domains and this may satisfy some unsatisfied constraints. For example, in the Cars Types table in Table 1, Car_Model and $DoorNo$ attributes are planned to be

a key but it is not satisfied as $sp\langle Car\_Model, DoorNo\rangle$ in the table. Removing two tuples can satisfy the key $sp\langle Car\_Model, DoorNo\rangle$, while adding one new tuple with a distinct door number value would satisfy the key. From some point of views, adding one tuple with new values is better than removing already existing two tuples. On the other hand, it is common that car models and the number of doors together determine the engine type, while the spFD $(Car\_Model, DoorNo) \rightarrow_{sp} Engine\_Type$ is violated in the table. So, adding a single tuple to the table with a new value for the attribute $DoorNo$ can satisfy the spFD $(Car\_Model, DoorNo) \rightarrow_{sp} Engine\_Type$, instead of taking off two tuples from the table.

Table 1: Types of Cars

Car_Model	Door No	Engine_Type
BMW	4 doors	$\perp$
BMW	$\perp$	e
Ford	$\perp$	V8
Ford	$\perp$	V6

2 Definitions

In a relation schema $R = \{A_1, A_2, \ldots A_n\}$, for an attribute A_i, let the domain of A_i be as $D_i = dom(A_i)$ for $i = 1,2,\ldots n$ and represent the set of values that are possible for each attribute $A_i \in R$. Then, for a subset $X \subseteq R$, the domain of X is $D_X = \prod_{\forall A_i \in K} D_i$.

An instance T over the relation R such that $T = (t_1,t_2,\ldots t_s)$ is a set of tuples such that every tuple represent a function $t : R \rightarrow \bigcup_{A_i \in R} dom(A_i)$ where $t[A_i] \in dom(A_i)$ for all A_i in R. For a set of tuples, several occurrences of the same tuple are allowed as we use the *bag semantics* concept. As the order of the tuples is not relevant, the *multiset of tuples* is considered as an instance. Let $t_r[X]$ represent the restriction of t_r to the attribute set $X \subset R$ for a tuple $t_r \in T$.

Let $\perp$ be a symbol in each attribute's domain that represents a missing value. Let V be a set of attributes, then t_r is V-total if $\forall A \in V$, $t_r[A] \neq \perp$. Furthermore, t_r is called a total tuple if it is R-total. Two tuples t_1 and t_2 are called *weakly similar* on $X \subseteq R$ written as $t_1[X] \sim_w t_2[X]$ defined by Köhler et.al. [11] if:

$$\forall A \in X \quad (t_1[A] = t_2[A] \text{ or } t_1[A] = \perp \text{ or } t_2[A] = \perp).$$

We use the notion $t_1 \sim_w t_2$ if t_1 and t_2 are weakly similar on each attribute in R.

In addition to that, t_1 and t_2 are called *strongly similar* on $X \subseteq R$ denoted by $t_1[X] \sim_s t_2[X]$ if

$$\forall A \in X \quad (t_1[A] = t_2[A] \neq \perp).$$

Let $T = (t_1, t_2, \ldots t_s)$ be an instance over the relation R, and $T' = (t'_1, t'_2, \ldots t'_s)$ is a *possible world* of T, such that T' is NULL-free and $t_i \sim_w t'_i \; \forall \; i = 1, 2, \ldots s$. This is by replacing each $\perp$ with a value, that is not $\perp$, from the attribute's domain for each tuple. The active domain of an attribute A_i is defined as the set of all the distinct values that appear on A_i except the NULL. Note that active domain was called *visible domain* in papers [3, 4, 5, 2].

Definition 1 *The* active domain *of an attribute A_i ($VD^T(A_i)$) is the set of all distinct values except $\perp$ that are already used by tuples in T:*

$$VD^T(A_i) = \{t[A_i] : t \in T\} \setminus \{\perp\} \textit{ for } A_i \in R.$$

The upper index T can be removed from the notation to simplify it if it is known which instance is considered.

A strongly possible world is achieved by using the active domain values for each occurrence of NULLs. following is a formal definition of the concept of *strongly possible world* that was introduced in [3].

Definition 2 *A possible world T' of T is called a* strongly possible world (spWorld) *if $t'[A_i] \in VD^T(A_i)$ for all $t' \in T'$ and $A_i \in R$.*

spWorlds are used to introduce *strongly possible keys (spKeys)* and *strongly possible functional dependencies (spFDs)* as follows.

Definition 3 *A strongly possible functional dependency, in notation $X \rightarrow_{sp} Y$, holds in table T over schema R if there exists a strongly possible world T' of T such that $T' \models X \rightarrow Y$. That is, for any $t'_1, t'_2 \in T'$ $t'_1[X] = t'_2[X]$ implies $t'_1[Y] = t'_2[Y]$. The set of attributes X is a strongly possible key, in notation $sp\langle X\rangle$, if there exists a strongly possible world T' of T where X is a key in T'. That is, for any $t'_1, t'_2 \in T'$ $t'_1[X] = t'_2[X]$ implies $t'_1 = t'_2$.*

If $T' = \{t'_1, t'_2, \ldots, t'_p\}$ is an spWorld of $T = \{t_1, t_2, \ldots, t_p\}$, then we say that t'_i is an *sp-extension* of t_i if $t_i \sim_w t'_i$. For a subset $X \subseteq R$, we say that $t'_i[X]$ is an *sp-extension* of t_i on X if $t_i \sim_w t'_i$ such that for each $A \in X$: $t'_i[A] \in VD(A)$.

3 Related work

g_3 measure was introduced by Kivinen et. al. in [12] for total tables. And Giannella et al. [10] introduced a measure to approximate the satisfaction of functional dependencies in a table. They introduced the approximation measure IFD and provided a compression with the two other approximation measures: g_3 (the minimum number of tuples that are required to be removed from the table to satisfy the dependency introduced in [12]) and τ (the probability of getting a true satisfaction guess of an FD introduced in [9]). Bounds were provided to find the differences and were applied to five datasets for analysis. It is shown that IFD and τ are more appropriate than g_3 when measuring the degree of knowledge for $X \rightarrow Y$ (applications like prediction, classification, and so on). On the other hand, g_3 measure is more

appropriate than IFD and τ when measuring the number of "violating" tuples in an FD.

In [15], Jef Wijsen studied some theoretical concepts in CQA (Consistent query answering), that is when a user sends queries to an inconsistent database regarding a set of constraints. They provided a database repairing by an acyclic binary relation $\leq_{db}$ on consistent database tables, such that $r_1 \leq_{db} r_2$ indicate that r_1 is at least as close to db as r_2. The minimum number of tuples that are required to be added and/or removed is one possible distance. Furthermore, the main concepts of database repairs and CQA were studied by Bertossi in [6]. J. Biskup and L. Wiese provide the preCQE algorithm that follows the formal properties of inference-proofness to find a solution for a given table in [7].

4 Approximation of strongly possible integrity constraints

Definition 4 *Attribute set K is an approximate strongly possible key of ratio a in table T, in notation $asp_a^- \langle K \rangle$, if there exists a subset S of the tuples T such that $T \setminus S$ satisfies $sp \langle K \rangle$, and $|S|/|T| \leq a$. The minimum a such that $asp_a^- \langle K \rangle$ holds is denoted by $g_3(K)$.*

The value of the measure $g_3(K)$ is between 0 and 1, where it is 0 when $sp \langle K \rangle$ holds in the table T (means it is not required to remove any tuples to satisfy the spKey). For this, we use the g_3 measure introduced in [12]. For example in Table 2, to satisfy the $sp\langle X \rangle$, we need to remove two out of four tuples as shown in Table 3, so that the g_3 is 0.5. The g_3 approximation measure for spKeys was introduced in [5]. This paper introduces the g_5 approximation measure for spKeys that is based on adding rather than removing tuples as in the following definition.

Definition 5 *Attribute set K is an add-approximate strongly possible key of ratio b in table T, in notation $asp_b^+ \langle K \rangle$, if there exists a set of tuples S such that the table TS satisfies $sp \langle K \rangle$, and $|S|/|T| \leq b$. The minimum b such that $asp_b^+ \langle K \rangle$ holds is denoted by $g_5(K)$.*

$g_5(K)$ is an approximation measure that is:

$$\frac{\text{minimum number of tuples to add}}{\text{total number of tuples}}$$

so that $sp \langle K \rangle$ holds. $g_5(K)$ measure has a value ranges between 0 and 1, where it equlas to 0 if $sp \langle K \rangle$ holds in T (means it is not required to remove any tuples to satisfy the spKey). For example in Table 2, to satisfy $sp\langle X \rangle$, it is enough to add one tuple as shown in Table 4, so that the g_5 is 0.25.

Definition 6 *For the attribute sets X and Y, $\sigma : X \rightarrow_{sp} Y$ is a remove-approximate strongly possible functional dependency of ratio a in a table T, in notation $T \models\approx_a^- X \rightarrow_{sp} Y$, if there exists a set of tuples S such that the table $T \setminus S \models X \rightarrow_{sp} Y$, and $|S|/|T| \leq a$. Then, $g_3(\sigma)$ is the smallest a such that $T \models\approx_a^- \sigma$ holds.*

Definition 7 *For the attribute sets X and Y, $\sigma : X \rightarrow_{sp} Y$ is an add-approximate strongly possible functional dependency of ratio b in a table T, in notation $T \models\approx_b^+ X \rightarrow_{sp} Y$, if there exists a set of tuples S such that the table $T \cup S \models X \rightarrow_{sp} Y$, and $|S|/|T| \leq b$. Then, $g_5(\sigma)$ is the smallest b such that $T \models\approx_b^+ \sigma$ holds.*

Let $U \subseteq T$ represent the set of the tuples that are required to be removed to have the spKey satisfied in T, in other words, $|U|$ tuples need to be removed. On other hand, adding only one tuple with new values may cause te satisfaction of the spKey in some tuples in U using the added new values for their `NULL`s. That means adding a number of tuples fewer than the those to remove can satisfy an spKey in the same table. For example, we either remove two tuples or add one to satisfy $sp \langle X \rangle$ in Table 2.

Table 2: Incomplete instance

X	
A_1	A_2
$\perp$	1
2	$\perp$
2	$\perp$
2	2

Table 3: Resulting table for ($asp_a^- \langle X \rangle$)

X	
A_1	A_2
$\perp$	1
2	2

4.1 Relation between g_3 and g_5 measures

Results together with their proofs of this subsection were reported in the conference volume [1], so the proofs are not

Table 4: The table after adding ($asp_b^+ \langle X \rangle$)

X	
A_1	A_2
$\perp$	1
2	$\perp$
2	$\perp$
2	2
3	3

included here, except for Theorem 1, which is shown for the sake of interested reader. The following Proposition is used to prove Proposition 2.

Proposition 1 *Let T be an instance over schema R and let $K \subseteq R$. If the K-total part of the table T satisfies the key $sp\langle K\rangle$, then there exists a minimum set of tuples U to be removed that are all non-K-total so that $T \setminus U$ satisfies $sp\langle K\rangle$.*

Proposition 2 *For any $K \subseteq R$ with $|K| \geq 2$, we have $g_3(K) \geq g_5(K)$.*

Apart form the previous inequality, the two measures are totally independent for spKeys.

Theorem 1 *Let $0 \leq \frac{p}{q} < 1$ be a rational number. Then there exist tables over schema $\{A_1, A_2\}$ with arbitrarily large number of rows, such that $g_3(\{A_1, A_2\}) - g_5(\{A_1, A_2\}) = \frac{p}{q}$.*

Proof: Table T is defined as follows.

$$T = \begin{array}{c} \\ b \\ \\ \\ \\ \\ x \\ \\ \\ \end{array} \begin{cases} 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & b \\ \bot & \bot \\ \bot & \bot \\ \vdots & \vdots \\ \bot & \bot \end{cases} \tag{1}$$

Clearly, $g_3(K) = \frac{x}{x+b}$. Let us assume that y tuples are needed to be added. The maximum number of active domain combinations is $(y+1)(y+b)$ obtained by adding tuples $(2, b+1), (3, b+2), \ldots, (y+1, y+b)$. This is enough to replace all tuples with NULLs if

$$(y+1)(y+b) \geq x + y + b. \tag{2}$$

On the other hand, $y-1$ added tuples are not enough, so

$$y(y-1+b) < x + y - 1 + b. \tag{3}$$

Since the total number of active domain combinations must be less than the tuples in the extended table. We have $\frac{p}{q} = g_3(K) - g_5(K) = \frac{x-y}{x+b}$ that is for some positive integer c we must have $cp = x - y$ and $cq = x + b$ if $gcd(p, q) = 1$. This can be rewritten as

$$y = x - cp\,; \quad y + b = c(q-p)\,; \quad b = cq - x\,; \quad x + y + b = y + cq. \tag{4}$$

Using (4) we obtain that (2) is equivalent with

$$y \geq \frac{cp}{c(q-p)-1}. \tag{5}$$

If c is large enough then $\lceil \frac{cp}{c(q-p)-1} \rceil = \lceil \frac{p}{q-p} \rceil$ so if $y = \lceil \frac{p}{q-p} \rceil$ is chosen then (5) and consequently (2) holds. On the other hand, (3) is equivalent to

$$y < \frac{cq-1}{c(q-p)-2}. \tag{6}$$

The right hand side of (6) tends to $\frac{q}{q-p}$ as c tends to infinity. Thus, for large enough c we have $\lfloor \frac{cq-1}{c(q-p)-2} \rfloor = \lfloor \frac{q}{q-p} \rfloor$. Thus, if

$$y = \lceil \frac{p}{q-p} \rceil \leq \lfloor \frac{q}{q-p} \rfloor \tag{7}$$

and $\frac{q}{q-p}$ is not an integer, then both (2) and (3) are satisfied for large enough c. Observe that $\frac{p}{q-p} + 1 = \frac{q}{q-p}$, thus (7) always holds. Also, if $\frac{q}{q-p}$ is indeed an integer, then we have strict inequality in (7) that implies (6) and consequently (3).

Unfortunately, the analogue of Proposition 1 is not true for spFDs, so the proof of the following theorem is quiet involved.

Theorem 2 *Let T be a table over schema R, $\sigma : X \to_{sp} Y$ for some $X, Y \subseteq R$. Then $g_3(\sigma) \geq g_5(\sigma)$.*

Theorem 3 can be proven by a construction similar to the proof of Theorem 1.

Theorem 3 *For any rational number $0 \leq \frac{p}{q} < 1$ there exists tables with an arbitrarily large number of rows and bounded number of columns that satisfy $g_3(\sigma) - g_5(\sigma) = \frac{p}{q}$ for $\sigma\colon X \to_{sp} Y$.*

4.2 Complexity problems

Definition 8 *The* SPKey problem *is the following.*
Input Table T over schema R and $K \subseteq R$.
Question Is it true that $T \models sp\langle K\rangle$?
The SPKeySystem problem *is the following.*
Input Table T over schema R and $\mathcal{K} \subseteq 2^R$.
Question Is it true that $T \models sp\langle \mathcal{K}\rangle$?
The SPFD problem *is the following.*
Input Table T over schema R and $X, Y \subseteq R$.
Question Is it true that $T \models X \to_{sp} Y$?

The following was shown in [4].

Theorem 4 *SPKey∈P, SPkeySystem and SPFD are NP-complete*

However, the approximation measures raise new, interesting algorithmic questions.

Definition 9 *The* SpKey-g3 *problem is the following.*
Input Table T over schema R, $K \subseteq R$ and $0 \leq q < 1$.
Question Is it true that $g_3(K) \leq q$ in table T?
The SpKey-g5 *problem is the following.*
Input Table T over schema R, $K \subseteq R$ and $0 \leq q < 1$.
Question Is it true that $g_5(K) \leq q$ in table T?

Proposition 3 *The decision problem SpKey-g5 is in P.*

Proof: Let us assume that tuples $s_i\colon i = 1, 2, \ldots, p$ over schema R are such that $T \cup \{s_1, s_2, \ldots s_p\}$ is optimal, so $g_5(K) = \frac{p}{m}$. Then clearly we may replace s_i by $s_i' = (z_i, z_i, \ldots, z_i)$ for all $i = 1, 2, \ldots, p$ where z_i's are pairwise distinct new values not appearing in the (extended) table $T \cup \{s_1, s_2, \ldots s_p\}$ so that $T \cup \{s_1', s_2', \ldots s_p'\} \models sp\langle K\rangle$.

Thus, if $g_5(K) \leq q$ is needed to be checked for a table T of m tuples, one may add $\lfloor q \cdot m \rfloor$ completely new tuples to obtain table T' and check whether $T' \models sp\langle K \rangle$ in polynomial time by Theorem 4.

Theorem 5 *Decision problem SpKey-g3 is in P.*

Proof: Let R be a relational schema and $K \subseteq R$. Furthermore, let T be an instance table over R that has some NULLs. Consider $T' = \{t' \in \Pi_{A \in K} VD^T(A) \colon \exists t \in T$ such that $t[K] \sim_w t'[K]\}$ and T' is total. Furthermore let the the bipartite graph $G = (T, T'; E)$ be the *K-extension graph of T* such that $\{t, t'\} \in E \iff t[K] \sim_w t'[K]$. So, finding a matching (if exists) of the graph G that covers T provides the tuples to be replaced in T to check if K is an spKey.

It was shown in [5] that the g_3 approximation measure for strongly possible keys satisfies

$$g_3(K) = \frac{|T| - \nu(G)}{|T|}.$$

where $\nu(G)$ denotes the maximum matching size in the K-extension graph G. However, the size of G is usually exponential function of the size of the input of the decision problem SpKey-g3, as T' is usually exponentially large.

In order to make our algorithm run in polynomial time we only generate part of T'. Let $T = \{t_1, t_2 \ldots t_m\}$ and $\ell(t_i) = |\{t^\star \in \Pi_{A \in K} VD^T(A) \colon t^\star \sim_w t_i[K]\}|$. Note that $\ell(t_i) = \prod_{A \colon t_i[A] = \perp} |VD^T(A)|$, hence these values can be calculated by scanning T once and using appropriate search tree data structures to hold values of active domains of each attribute. Sort tuples of T in non-decreasing $\ell(t_i)$ order, i.e. assume that $\ell(t_1) \leq \ell(t_2) \leq \ldots \leq \ell(t_m)$. Let $j = \max\{i \colon \ell(t_i) < i\}$ and $T_j = \{t_1, t_2, \ldots t_j\}$, furthermore $T_j^\star = \{t^\star : \exists t \in T_j : t^\star \sim_w t[K]\} \subseteq \Pi_{A \in K} VD^T(A)$. Note that $|T_j^\star| \leq \frac{1}{2} j(j-1)$. If $\forall i = 1, 2, \ldots, m \colon \ell(t_i) \geq i$, then define $j = 0$ and $T_j^\star = \emptyset$. Let $G^\star = (T_j, T_j^\star; E^\star)$ be the induced subgraph of G on the vertex set $T_j \cup T_j^\star$. Note that $|T_j^\star| \leq \frac{1}{2} j(j-1)$.
Claim $\nu(G) = \nu(G^\star) + |T \setminus T_j|$.
Proof of Claim: The inequality $\nu(G) \leq \nu(G^\star) + |T \setminus T_j|$ is straightforward. On the other hand, a matching of size $\nu(G^\star)$ in $G^\star$ can greedily be extended to the vertices in $|T \setminus T_j|$, as $t_i \in T \setminus T_j$ has at least i neighbours (which can be generated in polynomial time).

Thus it is enough to determine $\nu(G^\star)$ in order to calculate $g_3(K)$, and that can be done in polynomial time using Augmenting Path method [14].

Note that the proof above shows that the *exact value* of $g_3(K)$ can be determined in polynomial time. This gives the following corollary.

Definition 10 *The decision problem* SpKey-g3-equal-g5 *is defined as Input Table T over schema R, $K \subseteq R$.*
Question Is $g_3(K) = g_5(K)$?

Corollary 1 *The decision problem SpKey-g3-equal-g5 is in P.*

Example Let $R = \{A_1, A_2, A_3\}$, $K_1 = \{A_1, A_2\}, K_2 = \{A_2, A_3\}$.

$$T = \begin{array}{c|ccc} & A_1 & A_2 & A_3 \\ \hline t_1 & 1 & \perp & 1 \\ t_2 & 1 & 2 & 2 \\ t_3 & 2 & 1 & 1 \\ t_4 & 2 & 1 & 1 \end{array}$$

$T \setminus \{t_4\} \models sp\langle K_1 \rangle$ and $T \setminus \{t_4\} \models sp\langle K_2 \rangle$, but the sp-Worlds are different. In particular, this implies that for $\mathcal{K} = \{K_1, K_2\}$ we have $g_3(\mathcal{K}) > \max\{g_3(K) \colon K \in \mathcal{K}\}$ On the other hand, trivially $g_3(\mathcal{K}) \geq \max\{g_3(K) \colon K \in \mathcal{K}\}$ holds. This motivates the following definition.

Definition 11 *The problem* Max-g3 *defined as*
Input Table T over schema R, $\mathcal{K} \subseteq 2^R$.
Question Is $g_3(\mathcal{K}) = \max\{g_3(K) \colon K \in \mathcal{K}\}$?

Theorem 6 *Let Table T over schema R and $\mathcal{K} \subseteq 2^R$. The decision problem Max-g3 is NP-complete.*

Proof: The problem is in NP, a witness consists of a set of tuples U to be removed, an index $j \colon \frac{|U|}{|T|} = g_3(K_j)$, also an spWorld T' of $T \setminus U$ such that each K_i is a key in T'. Verifying the witness can be done in three steps.

1. $g_3(K_j) \not\leq \frac{|U|-1}{|T|}$ is checked in polynomial time using Theorem 5.
2. For all $i \neq j$ check that $g_3(K_i) \leq \frac{|U|}{|T|}$ using again Theorem 5.
3. Using standard database algorithms check that $\forall i \colon K_i$ is a key in T'.

On the other hand, the SPKeySystem problem can be Karp-reduced to the present question as follows. First check for each $K_i \in \mathcal{K}$ separately whether $sp\langle K_i \rangle$ holds, this can be done in polynomial time. If $\forall i \colon T \models sp\langle K_i \rangle$ then give $\mathcal{K}$ and T as input for Max-g3. It will answer Yes iff $T \models sp\langle \mathcal{K} \rangle$. However, if $\exists i \colon T \not\models sp\langle K_i \rangle$, then give the example above as input for Max-g3. Clearly both problems have No answer.

According to Theorem 4, it is NP-complete to decide whether a given SpFD holds in a table. Here we show that approximations are also hard.

Definition 12 *The* SPFD-g3 (SPFD-g5) *problems are defined as follows.*
Input A table T over schema R, $X, Y \subseteq R$, and positive rational number q.
Question Is $g_3(X \to_{sp} Y) \leq q$? ($g_5(X \to_{sp} Y) \leq q$?)

Theorem 7 *Both decision problems SPFD-g3 and SPFD-g5 are NP-complete.*

Proof: To show that SPFD-g3∈NP one may take a witness consisting of a subset $U \subset T$, an spWorld $T^\star$ of $T \setminus U$ such that $T^\star \models X \to Y$ and $|U|/|T| \leq q$. The validity

of the witness can easily be checked in polynomial time. Similarly, to show that SPFD-g5∈NP one may take a set of tuples S over R and an spWorld $T^\star$ of$T \cup S$ such that $T^\star \models X \to Y$ and $|S|/|T| \leq q$.

On the other hand, if $|T| = m$ and $q < 1/m$, then both SPFD-g3 and SPFD-g5 are equivalent with the original SPFD problem, since the smallest non-zero approximation measure is obtained if one tuple is needed to be deleted or added. According to Theorem 4, SPFD problem is NP-complete, thus so are SPFD-g3 and SPFD-g5.

Acknowledgement

The second author's research was partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–132696 and SNN–135643.

References

[1] Munqath Al-Atar and Attila Sali. Approximate keys and functional dependencies in incomplete databases with limited domains. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 147–167. Springer, 2022. doi: 10.1007/978-3-031-11321-5_9.

[2] Munqath Al-Atar and Attila Sali. Strongly possible functional dependencies for sql. *Acta Cybernetica*, 2022. doi: 10.1109/ACCESS.2022.3145678.

[3] Munqath Alattar and Attila Sali. Keys in relational databases with nulls and bounded domains. In *European Conference on Advances in Databases and Information Systems*, pages 33–50. Springer, 2019. doi: 10.1016/j.cose.2019.04.002.

[4] Munqath Alattar and Attila Sali. Functional dependencies in incomplete databases with limited domains. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 1–21. Springer, 2020. doi: 10.1016/j.is.2020.101522.

[5] Munqath Alattar and Attila Sali. Strongly possible keys for sql. *Journal on Data Semantics*, 9(2):85–99, 2020. doi: 10.1109/ACCESS.2020.3012345.

[6] Leopoldo Bertossi. Database repairs and consistent query answering: Origins and further developments. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 48–58, 2019. doi: 10.2200/S00974ED1V01Y201906DTM053.

[7] Joachim Biskup and Lena Wiese. A sound and complete model-generation procedure for consistent and confidentiality-preserving databases. *Theoretical Computer Science*, 412(31):4044–4072, 2011. doi: 10.1016/j.jcss.2010.06.004.

[8] Alireza Farhangfar, Lukasz A Kurgan, and Witold Pedrycz. A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(5):692–709, 2007. doi: 10.1007/s10115-006-0048-7.

[9] Leo A Goodman and William H Kruskal. Measures of association for cross classifications. *Measures of association for cross classifications*, pages 2–34, 1979. doi: 10.1080/01621459.1979.10481657.

[10] Chris Giannella and Edward Robertson. On approximation measures for functional dependencies. *Information Systems*, 29(6):483–507, 2004. doi: 10.1016/j.jcss.2004.03.001.

[11] Henning Köhler, Uwe Leck, Sebastian Link, and Xiaofang Zhou. Possible and certain keys for sql. *The VLDB Journal*, 25(4):571–596, 2016. doi: 10.1145/2882903.2915200.

[12] Jyrki Kivinen and Heikki Mannila. Approximate inference of functional dependencies from relations. *Theoretical Computer Science*, 149(1):129–149, 1995. doi: 10.14778/2153977.2154004.

[13] Witold Lipski Jr. On databases with incomplete information. *Journal of the ACM (JACM)*, 28(1):41–70, 1981. doi: 10.1145/322234.322239.

[14] László Lovász and Michael D Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[15] Jef Wijsen. Foundations of query answering on inconsistent databases. *ACM SIGMOD Record*, 48(3):6–16, 2019. doi: 10.2200/S00999ED1V01Y201910DTM055.

Online Bin Covering With Exact Parameter Advice

Andrej Brodnik[1], Bengt J. Nilsson[2], Gordana Vujovic[3]
[1]University of Ljubljana, Slovenia
[2]Malmö University, Sweden
[3]University of Ljubljana, Slovenia
E-mail: andrej.brodnik@fri.uni-lj.si, bengt.nilsson.TS@mau.se, gordana.vujovic@fri.uni-lj.si

Keywords: Online computation, Competitive analysis, Advice complexity, Bin covering

Received: April 14, 2023

We show an asymptotic $2/3$-competitive strategy for the bin covering problem using $O(b + \log n)$ bits of advice, where b is the number of bits used to encode a rational value and n is the length of the input sequence.

Povzetek: Raziskana je asimptotična 2/3-konkurentna strategija za problem pokrivanja binov, ki uporablja O(b + log n) bitov svetovanja, kjer je b število bitov za kodiranje racionalne vrednosti, n pa dolžina vhodnega zaporedja. Pristop temelji na izboljšavi strategije Dual Harmonic (DHk) s svetovanjem.

1 Introduction

In the bin covering problem, we are given a set of items of different sizes in the open range $]0, 1[$ and the goal is to find a maximum number of covered bins where a bin is covered if the sizes of items placed in it is at least 1. It has been shown that the bin covering problem is NP-hard [1, 2]. The covering problem has applications in various situations in business and in industry, from packing snack pieces into boxes so that each box contains at least its defined net weight, to such complex problems as redistribution tasks/items to a maximum number of factories/bins, all working at or beyond the minimal feasible level.

In the online version, items are delivered successively (one-by-one) and each item has to be packed, either in an existing bin or a new bin, before the next item arrives. The quality of online strategies is measured by their *competitive ratio*, the minimum ratio between the quality of the strategy's solution and that of an optimal one. The first known online strategy that has been proposed for the problem is *Dual Next Fit* (DNF), analogous to Next Fit for the bin packing problem. The competitive ratio of DNF is $1/2$ as proved by Assmann *et al*. [1]. Csirik and Totik [9] prove that no online algorithm can achieve a competitive ratio better than $1/2$. Additional lower bounds are provided by Balogh *et al*. [3].

The pure online framework is very restrictive in that it allows an all-powerful adversary to construct the input sequence in the worst possible way for the strategy making a competitive ratio better than $1/2$ unattainable.

Boyar *et al*. [5] look at bin covering using extra advice. If the input sequence consists of n items, they show that with $o(\log \log n)$ bits of advice, no strategy can have better competitive ratio than $1/2$. In addition, they show that a linear number of bits of advice is necessary to achieve competitive ratio greater than $15/16$. Their main result is a strategy with $O(\log \log n)$ bits of advice having competitive ratio $8/15$.

In order to provide exact advice values, e.g., an integer bounded by the number n, $\Omega(\log n)$ advice bits are required. As the strategy presented by Boyar *et al*. only provides approximate values of certain key parameters to achieve the $8/15$ competitive ratio, one can therefore ask if the competitive ratio can be improved if the oracle can give exact parameter values as advice. We answer this question affirmatively.

1.1 Our result

We show an asymptotic $2/3$-competitive strategy for the bin covering problem using $O(b + \log n)$ advice, where b is the number of bits used to encode a rational value in the input sequence and n is the length of the input sequence.

2 Preliminaries

The *online bin covering problem* we consider is, given an input sequence $\sigma = (v_1, v_2, \ldots)$, of rational values $v_i \in]0, 1[$, find the *maximum* number of unit sized bins that can be covered online with items from the input sequence σ.

We define the *load* of a bin B to be

$$\mathrm{ld}(B) \stackrel{\text{def}}{=} \sum_{v \in B} v. \tag{1}$$

We can similarly define the load of a sequence σ to be $\mathrm{ld}(\sigma) \stackrel{\text{def}}{=} \sum_{v \in \sigma} v$.

A *covering* is a partitioning of the items into bins $B_1, B_2, \ldots$ such that for each bin B_j

$$\mathrm{ld}(B_j) \geq 1 \tag{2}$$

and our objective is to find the maximum number of bins that satisfy Inequality (2). In contrast to the bin packing

problem, a strategy can open any number of bins at any time. However, only those that are filled to a load of at least 1 are counted in the solution.

We measure the quality of an online maximization strategy by its *competitive ratio*, the maximum bound R such that

$$\big|\mathsf{A}(\sigma)\big| \geq R \cdot \big|\mathsf{OPT}(\sigma)\big| - C, \tag{3}$$

for every possible input sequence σ, where $\mathsf{A}(\sigma)$ is the solution produced by the strategy A on σ, $\mathsf{OPT}(\sigma)$ is a solution on σ for which $|\mathsf{OPT}(\sigma)|$ is maximal, and C is some constant. If $C = 0$, we say that the competitive ratio is *strict* or *absolute*, otherwise it is *asymptotic*. The competitive ratio R is thus a positive real value ≤ 1, where equality to 1 implies that the strategy is (asymptotically) optimal.

Of particular interest is the Dual Next Fit strategy (DNF). DNF maintains one active bin B, and packs the items into B until it is covered. It then opens a new empty bin as the active bin and continues the process. As mentioned, Assmann *et al*. [1] prove that DNF has a competitive ratio of $1/2$ and Csirik and Totik [9] prove that no online algorithm can achieve a competitive ratio larger than $1/2$.

If the input sequence has some further structure, we can do slightly better as is shown in the next lemma that we will make extensive use of in the sequel.

Lemma 1 *The online strategy* DNF *for the bin covering problem on an input sequence σ_α where the items have weights bounded by $\alpha < 1$ has cost*

$$\big|\mathsf{DNF}(\sigma_\alpha)\big| > \frac{1}{1+\alpha}\big|\mathsf{OPT}(\sigma_\alpha)\big| - \frac{1}{1+\alpha}.$$

Proof: Assume that DNF opens $s+1$ bins when accessing the sequence σ_α, s of which are covered. Since every item has weight at most α, it means that each of the s covered bins are filled to a total weight of less than $1+\alpha$. A bin not obeying this limit would have been covered already before DNF places the last item in it, a contradiction. Thus the total load of the sequence σ_α is

$$(1+\alpha)s + 1 > \mathsf{ld}(\sigma_\alpha) \geq \big\lfloor \mathsf{ld}(\sigma_\alpha) \big\rfloor \geq \big|\mathsf{OPT}(\sigma_\alpha)\big|,$$

whereby $|\mathsf{DNF}(\sigma_\alpha)| = s > |\mathsf{OPT}(\sigma_\alpha)|/(1+\alpha) - 1/(1+\alpha)$ as claimed. □

Another strategy of interest is Dual Harmonic (DH_k), where the strategy subdivides the items by sizes into k groups,

$$]0, 1/k[,\ [1/k, 1/(k-1)[, \ldots, [1/3, 1/2[,\ [1/2, 1[,$$

and packs items in each group, maintaining k groups, according to DNF. Evidently, DH_k is at best $1/2$-competitive using the same argument as in Csirik and Totik [9].

In certain situations, the complete lack of information about future input is too restrictive. In a sense, the online strategy plays a game against an all-powerful adversary who can construct the input sequence in the worst possible manner. To alleviate the adversary's advantage, we consider the following *advice-on-tape* model [7]. An *oracle* has knowledge about both the strategy and the full input sequence from the adversary, it writes information encoded in binary on an *advice tape* of unbounded length. The strategy can read bits from the advice tape at any time, before or while the requests are released by the adversary. The *advice complexity* is the number of bits read from the advice tape by the strategy. Since the length of the advice bit string is not explicitly given, the oracle is unable to encode information into the length of the string, thereby requiring some mechanism to infer how many bits of advice the strategy should read at each step. This can be done with a self-delimiting encoding that extends the length of the bit string only by an additive lower order term [6].

A bit string s is encoded as $e(s) = u(s) \circ b(s) \circ s$ ($\circ$ denotes concatenation), where $b(s)$ is a binary encoding of the length of the string s and $u(s)$ consists of $\big|b(s)\big|$ ones followed by a single zero, thus indicating how many bits the strategy needs to read in order to obtain the length of the string s. The encoding has length at most $|e(s)| = |s| + 2\lceil \log(|s|+1) \rceil + 1$. We henceforth assume that all advice information is encoded in this way. An integer m can thus be encoded exactly using $O(\log m)$ bits and a rational value m_e/m_d, where m_e and m_d are integers can be encoded using $O(\log m_e + \log m_d)$ bits. If the rational value lies in the interval $[0, 1]$, then $m_e \leq m_d$ and the encoding can be made using $O(\log m_d)$ bits.

We will base our strategy on DH_k with added advice to improve on the competitive ratio, as do Boyar *et al*. [5].

3 Exact advice-based strategies for bin covering

Each item v in the input sequence corresponds to a rational value $0 < v < 1$, since any v above or equal to 1 will cover a bin and then the optimal solution can be assumed to place v alone in a bin to cover it. Also, values of size 0 could be placed in the first covered bin without loss of generality.

Fix an integer $k \geq 2$. We will subdivide the set of items into k subsets, such that $1/t \leq v < 1/(t-1)$ for each integer $2 \leq t \leq k$, the *t-items*, and items $v < 1/k$, the *small items*.

Consider a fixed optimal covering $\mathsf{OPT}(\sigma)$ for the input sequence σ. We can partition the solution $\mathsf{OPT}(\sigma)$ into groups, $\mathcal{G}_{t_1 t_2 \cdots t_j}$, where the index $t_1 t_2 \cdots t_j$, with $2 \leq t_1 \leq t_2 \leq \cdots \leq t_j \leq k$, denotes that each bin in group $\mathcal{G}_{t_1 t_2 \cdots t_j}$ contains one t_1-item, one t_2-item, etc, multiplicity denoting the number of times each item type occurs in the bin. The group of bins that are only covered by small items is denoted by $\mathcal{G}_S$.

We say that a bin in group $\mathcal{G}_{t_1 t_2 \cdots t_j}$ is *easy*, if $\sum_{t \in \{t_1, t_2, \ldots, t_j\}} 1/t \geq 1$ and we can assume without loss of generality that easy bins contain no small items. Furthermore, we assume that, if the bins in $\mathcal{G}_{t_1 t_2 \cdots t_j}$ are easy, then any bin group $\mathcal{G}_{t_1 t_2 \cdots t_{j+l}}$ is empty, if $t_1 t_2 \cdots t_j$ is a prefix of $t_1 t_2 \cdots t_{j+l}$, as any t_{j+i}-item, $1 \leq i \leq l$, in a bin B in $\mathcal{G}_{t_1 t_2 \cdots t_{j+l}}$ can be moved to other bins while B is still

covered. As an example, $\mathcal{G}_{22}$ are those bins that each contain two 2-items, so those bins are easy since two 2-items together guarantee that the bin is covered and if a bin in an optimal solution contains two 2-items, we assume that it does not contain any other items. As we noted, such items can be moved to other non-easy bins.

The non-easy bins may contain small items. Consider a bin in $\mathcal{G}_{23}$, if $k \geq 3$. It contains one 2-item and one 3-item. If the 2-item is 0.62 and the 3-item is 0.41, then the items cover the bin but this is not necessarily guaranteed since, if the two items are 0.55 and 0.35, there must be small items in the bin for it to be covered.

We also say that a bin in $\mathcal{G}_{t_1t_2\cdots t_j}$ is a *gap bin*, if $\sum_{t\in\{t_1,t_2,\ldots,t_j\}} 1/(t-1) < 1$, as each of these bins must contain small items to the total amount of more than $1-\sum_{t\in\{t_1,t_2,\ldots,t_j\}} 1/(t-1)$ to be covered. For example, the bins in $\mathcal{G}_3$ are gap bins since they all have to contain small items to a total amount of more than $1/2$ to be covered.

The size of the optimal solution is given by

$$|\mathsf{OPT}(\sigma)| = \sum_{\forall t_1t_2\cdots t_j} |\mathcal{G}_{t_1t_2\cdots t_j}| + |\mathcal{G}_S|, \tag{4}$$

for all valid index combinations $t_1t_2\cdots t_j$.

We modify DH_k to operate on advice and describe this strategy, denoted $\mathsf{DH}_k^{b+\lg n}$, dependent on the parameter k, the number of item types used to partition the items into. The superscript indicates the amount of advice that the strategy admits. Let $x_1,\ldots,x_n$, $n=|\sigma|$, be an ordering of the items in σ, such that $x_i \geq x_{i+1}$, for $1 \leq i < |\sigma|$. The oracle provides the strategy with an integer m and the value x_m through a self-delimiting encoding as previously described.

The objective of providing the parameters m and x_m is to give the strategy enough information to emulate the construction of the bin group $\mathcal{G}_2$ in an optimal solution, independently of the ordering in which items of the input sequence occur. The value of m is balanced by the size $|\mathcal{G}_2|$, also designating the number of 2-items packed alone but together with small items in the bins of $\mathcal{G}_2$, and the amount of small items present in the bins of $\mathcal{G}_2$. In the optimal solution, each of these bins could be covered to exactly the value 1 but any online strategy may have to overfill by an amount $< 1/k$.

The oracle, given the knowledge of the input sequence and the strategy $\mathsf{DH}_k^{b+\lg n}$, can determine the best value for m by emulating $\mathsf{DH}_k^{b+\lg n}$ on the input sequence, for each integer $0 \leq m \leq |\mathcal{G}_2|$, if it knows an optimal solution. If it does not, it counts the number of 2-items, n_2, in the input sequence and tries all values between $0 \leq m \leq n_2$ and reports an m value for which $\mathsf{DH}_k^{b+\lg n}$ delivers as large a solution as possible. Since $n_2 \geq |\mathcal{G}_2|$ the best solution is no worse than if m is restricted to be $\leq |\mathcal{G}_2|$. Given a specific integer m, the oracle can provide the value x_m using the standard selection algorithm [4].

The strategy $\mathsf{DH}_k^{b+\lg n}$ initially reads the parameters m and x_m and opens m bins that we call *critical bins* and that will each be covered with one of the m largest 2-items of the input sequence σ together with small items. Initially, each critical bin is assumed to have a *virtual load* of x_m. When an item of size $\geq x_m$ is placed in a critical bin, its virtual load is increased to the actual value of the item. The strategy further opens a *t-bin* for every item type $t \in \{2,\ldots,k\}$, and a *small bin* for the small items. As the next item v of the input sequence arrives, it is handled as follows:

1. if $1/2 \leq x_m \leq v$, place v in the next critical bin that does not yet contain a 2-item and update the virtual load of the critical bin. If all critical bins already contain 2-items, go to the next step,
2. if $1/2 \leq v < x_m$ or $x_m \leq v$ and all critical bins contain a 2-item, place v in the 2-bin using DNF. If the bin becomes covered, close it and open a new 2-bin,
3. if $1/k \leq v < 1/2$ is a t-item, place v in the corresponding t-bin using DNF. If the bin becomes covered, close it and open a new t-bin,
4. if $v < 1/k$ is small, place v in the next critical bin that does not contain small items up to a virtual load of at least 1 and update the virtual load of this critical bin. If all critical bins are filled up to a virtual load of 1, place v in the small bin using DNF. If the small bin becomes covered, close it and open a new small bin.

Consider the following example input sequence

$$\begin{aligned}\sigma = \big(&0.25, 0.8, 0.72, 0.2, 0.9, 0.45, 0.51, 0.67, 0.45, 0.6,\\ &0.42, 0.55, 0.53, 0.28, 0.11, 0.15, 0.52, 0.15, 0.51,\\ &0.41, 0.15, 0.35, 0.1, 0.35, 0.3, 0.3, 0.4, 0.18\big),\end{aligned}$$

taken from [5] and slightly modified. Figure 1 shows the optimal solution and the $\mathsf{DH}_3^{b+\lg n}$ solution on the sequence σ, for $k=3$, given that the oracle provides the values $m=2$ and $x_m = 0.8$.

We prove the following intermediate result to give the idea of the analysis for the more extended version in Theorem 2.

Theorem 1 *Assume that the strategy $\mathsf{DH}_2^{b+\lg n}$ has access to the exact values of m and x_m, then it has asymptotic competitive ratio*

$$\big|\mathsf{DH}_2^{b+\lg n}(\sigma)\big| \geq \frac{3}{5}\big|\mathsf{OPT}(\sigma)\big| - \frac{19}{15},$$

for serving any sequence σ of size n.

Proof: Since $k=2$, our strategy uses only two item types, 2-items and small items. The optimal solution consist of three bin groups with $|\mathsf{OPT}(\sigma)| = |\mathcal{G}_{22}|+|\mathcal{G}_2|+|\mathcal{G}_S|$. Note that the number of 2-items in σ is exactly

$$T_2 = 2|\mathcal{G}_{22}| + |\mathcal{G}_2|. \tag{5}$$

Consider now some arbitrary set of covered bins $\mathcal{G}$, where each bin only contains small items. Assume that these bins have a total load of $S = \sum_{B\in\mathcal{G}} \mathsf{ld}(B) \geq |\mathcal{G}|$ and that the input sequence restricted to these small items is σ_S. From Lemma 1 we have that

$$\big|\mathsf{DNF}(\sigma_S)\big| > \frac{2}{3}S - \frac{2}{3} \geq \frac{2}{3}|\mathcal{G}| - \frac{2}{3} \tag{6}$$

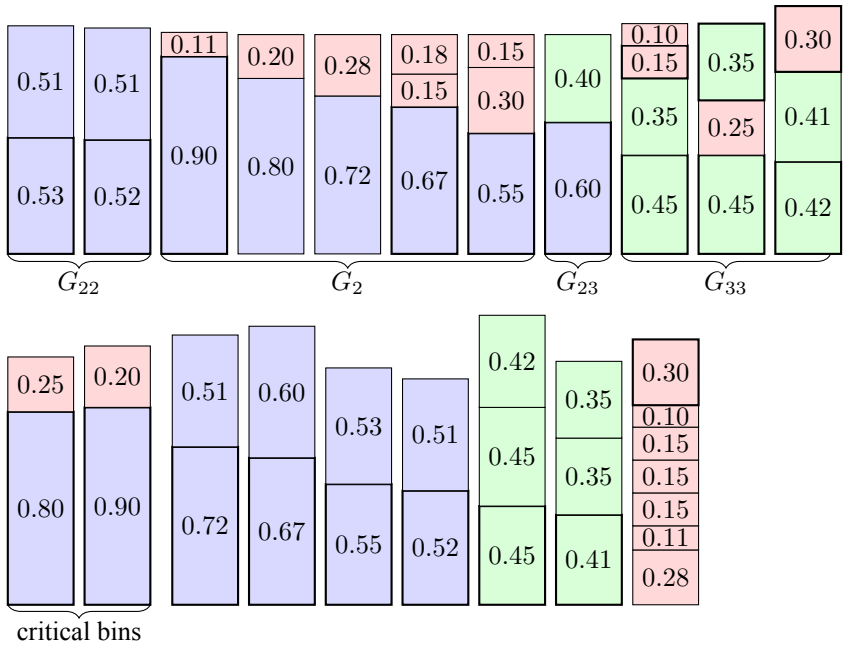


Figure 1: (top) An optimal covering for instance σ with 11 bins. (bottom) A $\mathsf{DH}_3^{b+\lg n}$ covering with 9 bins.

We let $S_S \stackrel{\text{def}}{=} \sum_{B\in\mathcal{G}_S} \mathrm{ld}(B)$ be the total load of the small items covering the bins in $\mathcal{G}_S$.

Next, we analyze the competitive ratio of the critical bins. Consider a bin B in $\mathcal{G}_2$. It contains precisely one 2-item having some weight $\geq 1/2$. Assume that we have sorted the bins in $\mathcal{G}_2$ in order of decreasing weight of its 2-item, i.e., we have an ordering of the bins $B_1,\ldots,B_{|\mathcal{G}_2|}$ such that the weight if the 2-item in B_i is w_i, $1\leq i\leq|\mathcal{G}_2|$ and $w_i \geq w_{i+1}$ for $1\leq i\leq|\mathcal{G}_2|-1$; see Figure 2. Let $u_i = \mathrm{ld}(B_i) - w_i$ be the weight of the small items in B_i and let $S_2 = \sum_{1\leq i\leq|\mathcal{G}_2|} u_i$ be the total load of the small items covering the bins in $\mathcal{G}_2$. It is clear that $S_2 \geq (|\mathcal{G}_2|-m)(1-w_m)$ for arbitrary choice of $m\leq|\mathcal{G}_2|$ since $B_{m+1},\ldots,B_{|\mathcal{G}_2|}$, each contains at least $1-w_m$ amount of small items.

Let C_i, $1\leq i\leq m$, be the critical bins opened by our strategy in the order they are constructed. Let $a_i \geq x_m \geq w_m$ denote the weight of the 2-item in C_i, let z_i be the weight of the last small item placed in C_i by our strategy, and let $y_i = \mathrm{ld}(C_i) - a_i - z_i$ be the weight of the remaining small items in C_i. By construction, $y_i \leq 1-w_m$ for each $1\leq i\leq m$; see Figure 2.

Let $\mathcal{I}_2(m)\subseteq\{1,\ldots,m\}$ be the set of indices i such that the last small element (of weight z_i) that was placed in critical bin C_i was placed by the optimal solution in a bin in $\mathcal{G}_2$. Similarly, let $\mathcal{I}_S(m) = \{1,\ldots,m\}\setminus\mathcal{I}_2(m)$ be the set of remaining indices. The possible values of m range between $0\leq m\leq m^+ = \lfloor(|\mathcal{G}_2| - |\mathcal{I}_2(m^+)|)/2\rfloor$, where m^+ is the largest integer such that $2m^+ + |\mathcal{I}_2(m^+)| \leq |\mathcal{G}_2|$, since the strategy needs to guarantee that it can cover all the critical bins.

The oracle can ascertain the value m^+ by emulating the strategy over all possible integer values and it reveals the values $m = m^+ = \lfloor(|\mathcal{G}_2| - |\mathcal{I}_2(m^+)|)/2\rfloor$ and $x_m = x_{m^+}$, the size of the m^{th} largest item in the input sequence σ to the strategy. The strategy constructs m critical bins, $\lfloor(T_2-m)/2\rfloor$ 2-bins and some bins corresponding to the amount of unused small items, giving us

$$
\begin{aligned}
\left|\mathsf{DH}_2^{b+\lg n}(\sigma)\right| &> m + \left\lfloor\frac{T_2-m}{2}\right\rfloor + \frac{2}{3}\Bigg(S_S + S_2 \\
&\quad - \left(\sum_{i=1}^{m} y_i + z_i\right)\Bigg) - \frac{2}{3} \\
&\geq m + \frac{T_2-m}{2} - \frac{1}{2} + \frac{2}{3}\Bigg(S_S + S_2 - \sum_{i=1}^{m}(1-w_m) \\
&\quad - \sum_{i=1}^{m} z_i\Bigg) - \frac{2}{3} \\
&= \frac{m}{2} + \frac{T_2}{2} + \frac{2}{3}\left(S_S - \sum_{i\in\mathcal{I}_S(m)} z_i\right) \\
&\quad + \underbrace{\frac{2}{3}\left(S_2 - \sum_{i=1}^{m}(1-w_m) - \sum_{i\in\mathcal{I}_2(m)} z_i\right)}_{\geq 0} - \frac{7}{6} \\
&\geq \frac{m}{2} + \frac{T_2}{2} + \frac{2}{3}\left(S_S - \sum_{i\in\mathcal{I}_S(m)} z_i\right) - \frac{7}{6} \\
&\geq \frac{m}{2} + \frac{T_2}{2} + \frac{3}{5}\left(S_S - \sum_{i\in\mathcal{I}_S(m)} z_i\right) - \frac{7}{6} \\
&> \frac{m}{2} + \frac{T_2}{2} + \frac{3S_S}{5} - \frac{3|\mathcal{I}_S(m)|}{10} - \frac{7}{6} \\
&= \frac{m}{5} + \frac{T_2}{2} + \frac{3S_S}{5} + \frac{3|\mathcal{I}_2(m)|}{10} - \frac{7}{6} \\
&= \frac{\lfloor(|\mathcal{G}_2| - |\mathcal{I}_2(m)|)/2\rfloor}{5} + \frac{2|\mathcal{G}_{22}| + |\mathcal{G}_2|}{2} + \frac{3S_S}{5} \\
&\quad + \frac{3|\mathcal{I}_2(m)|}{10} - \frac{7}{6} \;\geq\; |\mathcal{G}_{22}| + \frac{3|\mathcal{G}_2|}{5} + \frac{3|\mathcal{G}_S|}{5} - \frac{19}{15}
\end{aligned}
$$

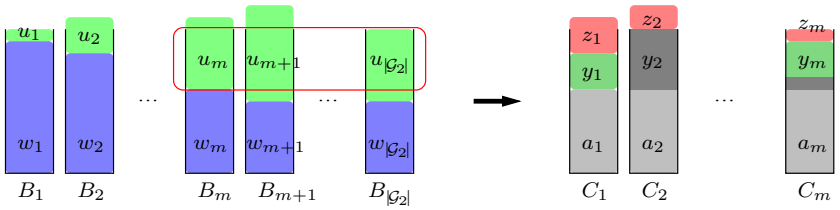


Figure 2: The critical bins and their relationship to the $\mathcal{G}_2$-bins in the optimal covering. In the $\mathcal{G}_2$-bins, blue are 2-items and light green are small items. In the critical bins, red represents the last small item placed in the bin, dark green are the remaining small items, grey items are the 2-items, and dark grey represents the overlap between the virtual and actual load of the 2-item.

bins, by applying Equality (5) in the last inequality. In addition, we use that each $z_i < 1/2$, that $S_S \geq \sum_{i \in \mathcal{I}_S(m)} z_i$, that critical bin C_i can be covered by a 2-item of size at least w_m plus the small items from a bin among $B_{m+1}, \ldots, B_{|\mathcal{G}_2|}$ and possibly one extra small item from a bin among $\{B_{m+1}, \ldots, B_{|\mathcal{G}_2|}\} \cup \mathcal{G}_S$; see Figure 2. Furthermore, $m = |\mathcal{I}_2(m)| + |\mathcal{I}_S(m)|$, for any m. The competitive ratio is the smallest coefficient of any of the terms corresponding to bin groups, since an adversary can ensure that the groups with larger coefficient contain no bins. This gives a competitive ratio of $3/5 = 0.6$. □

We have presented the proof for two item types to illustrate the general idea of the next result. For two item types, the number of bin groups is three, whereas for four item types, the number of bin groups increases to 20, but the proof steps are exactly the same. For completeness we mention that $|\mathsf{DH}_3^{b+\lg n}(\sigma)| \geq 9|\mathsf{OPT}(\sigma)|/14 - 97/42$, where $9/14 \approx 0.64285\ldots$.

Lemma 2 *Assume that the strategy $\mathsf{DH}_4^{b+\lg n}$ has access to the exact values of m and x_m, then it has asymptotic competitive ratio*

$$\left|\mathsf{DH}_4^{b+\lg n}(\sigma)\right| \geq \frac{2}{3}\left|\mathsf{OPT}(\sigma)\right| - \frac{173}{60}$$

for serving any sequence σ of size n.

Proof: The number of t-items, for $t = 2$, 3, and 4, in the instance is

$$\begin{aligned} T_2 &= |\mathcal{G}_2| + 2|\mathcal{G}_{22}| + |\mathcal{G}_{23}| + |\mathcal{G}_{24}| + |\mathcal{G}_{233}| + \\ &\quad + |\mathcal{G}_{234}| + |\mathcal{G}_{244}|, \qquad (7) \\ T_3 &= |\mathcal{G}_3| + |\mathcal{G}_{23}| + 2|\mathcal{G}_{33}| + |\mathcal{G}_{34}| + 2|\mathcal{G}_{233}| + \\ &\quad + |\mathcal{G}_{234}| + 3|\mathcal{G}_{333}| + 2|\mathcal{G}_{334}| + |\mathcal{G}_{344}| + \\ &\quad + 2|\mathcal{G}_{3344}|, + |\mathcal{G}_{3444}|, \qquad (8) \\ T_4 &= |\mathcal{G}_4| + |\mathcal{G}_{24}| + |\mathcal{G}_{34}| + 2|\mathcal{G}_{44}| + |\mathcal{G}_{234}| + \\ &\quad + 2|\mathcal{G}_{244}| + |\mathcal{G}_{334}| + 2|\mathcal{G}_{344}| + 3|\mathcal{G}_{444}| + \\ &\quad + 2|\mathcal{G}_{3344}| + 3|\mathcal{G}_{3444}| + 4|\mathcal{G}_{4444}|. \qquad (9) \end{aligned}$$

For each non-easy bin group $\mathcal{G}_2, \ldots, \mathcal{G}_{444}$ (there are eight of them), let $S_{t_1 \cdots t_4}$ denote the weight of the small items that the optimum solution packs in the bins of group $\mathcal{G}_{t_1 \cdots t_4}$. In addition, we denote by $S_S = \sum_{B \in \mathcal{G}_S} \mathrm{ld}(B)$ the total load of the small items covering the bins in $\mathcal{G}_S$.

As in the previous proof, we first consider some arbitrary set of covered bins $\mathcal{G}$, where each bin only contains small items. Assume that these bins have a total load of $S = \sum_{B \in \mathcal{G}} \mathrm{ld}(B) \geq |\mathcal{G}|$ and that the input sequence restricted to these small items is σ_S. From Lemma 1 we have that

$$\left|\mathsf{DNF}(\sigma_S)\right| > \frac{4}{5}S - \frac{4}{5} \geq \frac{4}{5}|\mathcal{G}| - \frac{4}{5} \qquad (10)$$

We can analyze the competitive ratio of the critical bins exactly as in the previous proof. First consider a decreasing ordering of the bins $B_1, \ldots, B_{|\mathcal{G}_2|}$ in $\mathcal{G}_2$ by the weight of their 2-item, w_i. We let $u_i = \mathrm{ld}(B_i) - w_i$ be the weight of the small items in B_i, whereby $S_2 \geq \big(|\mathcal{G}_2| - m\big) \cdot \big(1 - w_m\big)$, for arbitrary choice of $m \leq |\mathcal{G}_2|$. Each critical bin, C_i, $1 \leq i \leq m$, contains one 2-item of weight a_i, a small item of weight z_i that was the last small item placed in C_i by our strategy, and small items to the weight of $y_i = \mathrm{ld}(C_i) - a_i - z_i$. Again, $y_i \leq 1 - w_m$, for each $1 \leq i \leq m$.

Consider next the gap bins in the optimal solution. These are the bins in groups $\mathcal{G}_3$, $\mathcal{G}_4$, $\mathcal{G}_{34}$, and $\mathcal{G}_{44}$. Each bin in these groups is guaranteed to have small items to the amount of at least $1/2$, $2/3$, $1/6$, and $1/3$, respectively. Thus, for each of those groups we have $S_3 \geq |\mathcal{G}_3|/2$, $S_4 \geq 2|\mathcal{G}_4|/3$, $S_{34} \geq |\mathcal{G}_{34}|/6$, and $S_{44} \geq |\mathcal{G}_{44}|/3$.

For each group of non-easy bins $\mathcal{G}_2, \ldots, \mathcal{G}_{444}$, let $\mathcal{I}_{t_1 \cdots t_4}(m) \subseteq \{1, \ldots, m\}$ be the set of indices i such that the last small element (of weight z_i) that was placed in critical bin C_i was placed by the optimal solution in a bin from bin group $\mathcal{G}_{t_1 \cdots t_4}$. Also, let $\mathcal{I}_S(m) = \{1, \ldots, m\} \setminus \big(\bigcup_{t_1 \cdots t_4 \notin Easy} \mathcal{I}_{t_1 \cdots t_4}(m)\big)$ be the set of remaining indices. As before, the possible values of m range between $0 \leq m \leq m^+ = \big\lfloor(|\mathcal{G}_2| - |\mathcal{I}_2(m^+)|)/2\big\rfloor$, where m^+ is the largest integer such that $2m^+ + |\mathcal{I}_2(m^+)| \leq |\mathcal{G}_2|$, since the strategy needs to guarantee that it can cover all the critical bins. The oracle ascertains the maximum integer m^+ and reveals the values $m = m^+ = \big\lfloor(|\mathcal{G}_2| - |\mathcal{I}_2(m^+)|)/2\big\rfloor$ and $x_m = x_{m^+}$, so our strategy constructs m critical bins, $\big\lfloor(T_2 - m)/2\big\rfloor$ 2-bins, $\big\lfloor T_3/3\big\rfloor$ 3-bins, $\big\lfloor T_4/4\big\rfloor$ 4-bins, and bins corresponding to the amount of unused small items, giving us

$$
\begin{aligned}
\left|\mathsf{DH}_4^{b+\lg n}(\sigma)\right| &> m + \left\lfloor \frac{T_2 - m}{2} \right\rfloor + \left\lfloor \frac{T_3}{3} \right\rfloor + \left\lfloor \frac{T_4}{4} \right\rfloor \\
&\quad + \frac{4}{5}\Bigg(S_S + S_2 + S_3 + S_4 + S_{34} + S_{44} \\
&\quad - \Bigg(\sum_{i=1}^{m} y_i + z_i \Bigg) \Bigg) - \frac{4}{5} \\
&\geq \frac{m}{2} + \frac{T_2}{2} + \frac{T_3}{3} + \frac{T_4}{4} + \frac{4}{5}\Bigg(S_S - \sum_{i \in \mathcal{I}_S(m)} z_i \Bigg) \\
&\quad + \frac{4}{5}\Bigg(S_3 - \sum_{i \in \mathcal{I}_3(m)} z_i \Bigg) + \frac{4}{5}\Bigg(S_4 - \sum_{i \in \mathcal{I}_4(m)} z_i \Bigg) \\
&\quad + \frac{4}{5}\Bigg(S_{34} - \sum_{i \in \mathcal{I}_{34}(m)} z_i \Bigg) + \frac{4}{5}\Bigg(S_{44} - \sum_{i \in \mathcal{I}_{44}(m)} z_i \Bigg) \\
&\quad + \underbrace{\frac{4}{5}\Bigg(S_2 - \sum_{i=1}^{m}(1 - w_m) - \sum_{i \in \mathcal{I}_2(m)} z_i \Bigg)}_{\geq 0} - \frac{163}{60} \\
&\geq \frac{m}{2} + \frac{T_2}{2} + \frac{T_3}{3} + \frac{T_4}{4} + \frac{2}{3}\Bigg(S_S - \sum_{i \in \mathcal{I}_S(m)} z_i \Bigg) \\
&\quad + \frac{2}{3}\Bigg(S_3 - \sum_{i \in \mathcal{I}_3(m)} z_i \Bigg) + \frac{5}{8}\Bigg(S_4 - \sum_{i \in \mathcal{I}_4(m)} z_i \Bigg) \\
&\quad + \frac{1}{2}\Bigg(S_{34} - \sum_{i \in \mathcal{I}_{34}(m)} z_i \Bigg) + \frac{1}{2}\Bigg(S_{44} - \sum_{i \in \mathcal{I}_{44}(m)} z_i \Bigg) - \frac{163}{60} \\
&\geq \frac{m}{2} + \frac{T_2}{2} + \frac{T_3}{3} + \frac{T_4}{4} + \frac{2S_S}{3} - \frac{|\mathcal{I}_S(m)|}{6} \\
&\quad + \frac{2S_3}{3} - \frac{|\mathcal{I}_3(m)|}{6} + \frac{5S_4}{8} - \frac{5|\mathcal{I}_4(m)|}{32} \\
&\quad + \frac{S_{34}}{2} - \frac{|\mathcal{I}_{34}(m)|}{8} + \frac{S_{44}}{2} - \frac{|\mathcal{I}_{44}(m)|}{8} - \frac{163}{60} \\
&\geq \frac{m}{2} + \frac{T_2}{2} + \frac{T_3}{3} + \frac{T_4}{4} + \frac{2S_S}{3} - \frac{m}{6} + \frac{|\mathcal{I}_2(m)|}{6} \\
&\quad + \frac{|\mathcal{I}_3(m)|}{6} + \frac{|\mathcal{I}_4(m)|}{6} + \frac{|\mathcal{I}_{34}(m)|}{6} + \frac{|\mathcal{I}_{44}(m)|}{6} \\
&\quad + \frac{2S_3}{3} - \frac{|\mathcal{I}_3(m)|}{6} + \frac{5S_4}{8} - \frac{5|\mathcal{I}_4(m)|}{32} \\
&\quad + \frac{S_{34}}{2} - \frac{|\mathcal{I}_{34}(m)|}{8} + \frac{S_{44}}{2} - \frac{|\mathcal{I}_{44}(m)|}{8} - \frac{163}{60} \\
&= \frac{2|\mathcal{G}_2|}{3} + \frac{2|\mathcal{G}_3|}{3} + \frac{2|\mathcal{G}_4|}{3} + \frac{2|\mathcal{G}_{33}|}{3} + \frac{2|\mathcal{G}_{34}|}{3} + \frac{2|\mathcal{G}_{44}|}{3} \\
&\quad + \frac{2|\mathcal{G}_S|}{3} + \frac{3|\mathcal{G}_{24}|}{4} + \frac{3|\mathcal{G}_{444}|}{4} + \frac{5|\mathcal{G}_{23}|}{6} + \frac{5|\mathcal{G}_{344}|}{6} \\
&\quad + \frac{11|\mathcal{G}_{334}|}{12} + |\mathcal{G}_{22}| + |\mathcal{G}_{244}| + |\mathcal{G}_{333}| + |\mathcal{G}_{4444}| \\
&\quad + \frac{13|\mathcal{G}_{234}|}{12} + \frac{13|\mathcal{G}_{3444}|}{12} + \frac{7|\mathcal{G}_{233}|}{6} + \frac{7|\mathcal{G}_{3344}|}{6} - \frac{173}{60} \\
&\geq \frac{2}{3}|\mathsf{OPT}(\sigma)| - \frac{173}{60}
\end{aligned}
$$

bins, by applying Equalities (7)–(9) in the second to last inequality above. We further use that each $z_i < 1/4$, that $S_S \geq \sum_{i \in \mathcal{I}_S(m)} z_i$ and $S_{t_1 \cdots t_4} \geq \sum_{i \in \mathcal{I}_{t_1 \cdots t_4}(m)} z_i$, for each bin group $\mathcal{G}_{t_1 \cdots t_4}$. Every critical bin C_i can be covered by a large item of size at least w_m plus the small items from a bin among the last bins $B_{m+1}, \ldots, B_{|\mathcal{G}_2|}$ in $\mathcal{G}_2$ and possibly one extra small item from a non-easy bin in the optimal solution; see Figure 2. Furthermore, $m = |\mathcal{I}_S(m)| + \sum_{t_1 \cdots t_4} |\mathcal{I}_{t_1 \cdots t_4}(m)|$, for any m. The competitive ratio is the smallest coefficient of any of the terms corresponding to bin groups, since an adversary can ensure that the groups with larger coefficient contain no bins. This gives a competitive ratio of $2/3 \approx 0.6666\ldots$. □

The two advice values $m \leq n$ and x_m can be represented by $O(\log n)$ bits and $O(b)$ bits respectively, where b is the number of bits required to represent the integer denominator of the rational value x_m, since $x_m < 1$. We have the following immediate theorem.

Theorem 2 *The strategy* $\mathsf{DH}_4^{b+\lg n}$ *receives* $O(b + \log n)$ *bits of advice and has asymptotic competitive ratio*

$$\left|\mathsf{DH}_4^{b+\lg n}(\sigma)\right| \geq \frac{2}{3}\left|\mathsf{OPT}(\sigma)\right| - \frac{173}{60}$$

for serving any sequence σ *of size* n*, where* b *is the number of bits required to represent any rational value in* σ*.*

By approximating the advice value m using $O(\log \log n)$ of the most significant bits (using a modified self-delimiting encoding), a slight variation of the analysis above shows that a variant of $\mathsf{DH}_4^{b+\lg n}$ that receives $O(b + \log \log n)$ bits of advice still has asymptotic competitive ratio $2/3 - O(1/\log n)$, thus the exact value of m has very little impact on the competitive ratio.

3.1 Tightness

One could venture to think that strategy $\mathsf{DH}_k^{b+\lg n}$, for $k > 4$, would give improved competitive ratio, or even that extending the strategy with more sets of critical bins could improve it further. However, this is not possible, since an adversary can simply provide an instance where all bin groups except $\mathcal{G}_2$ in an optimal solution are empty. Thus, the instance consists of only 2-items and small items. Indeed, such instances form the basis for the lower bound proof for DNF [1, 2]. Any critical bin-based strategy must solve this instance and chooses some value for m, the number of critical bins to open. Even if the adversary provides all the small items first and the 2-items last, the strategy will cover $m + \lfloor(|\mathcal{G}_2| - m)/2\rfloor$ bins as long as the strategy can guarantee that all m critical bins are covered with one 2-item and some small items. Since the index set is $\mathcal{I}_2(m) = \{1, \ldots, m\}$, for all m, the maximum occurs for $m = \lfloor|\mathcal{G}_2|/3\rfloor$. The strategy covers at most

$$
\begin{aligned}
m + \left\lfloor \frac{|\mathcal{G}_2| - m}{2} \right\rfloor &= \left\lfloor \frac{|\mathcal{G}_2|}{3} \right\rfloor + \left\lfloor \frac{|\mathcal{G}_2| - \lfloor|\mathcal{G}_2|/3\rfloor}{2} \right\rfloor \\
&\leq \frac{|\mathcal{G}_2|}{3} + \frac{|\mathcal{G}_2|}{3} + \frac{1}{3} \leq \frac{2}{3}|\mathsf{OPT}(\sigma)| + \frac{1}{3}
\end{aligned}
$$

bins, proving that our analysis in Lemma 2 is asymptotically tight.

References

[1] Susan Fera Assmann, David S. Johnson, Daniel J. Kleitman, and Joseph Y-T. Leung. On

a dual version of the one-dimensional bin packing problem. *Journal of Algorithms*, 5(4):502–525, 1984. https://doi.org/10.1016/0196-6774(84)90004-X

[2] Susan Fera Assmann. *Problems in discrete applied mathematics*. PhD thesis, Massachusetts Institute of Technology, 1983. https://hdl.handle.net/1721.1/121905

[3] János Balogh, Leah Epstein, and Asaf Levin. Lower bounds for online bin covering-type problems. *Journal of Scheduling*, 22(4):487–497, 2019. https://doi.org/10.1007/s10951-018-0590-0

[4] Manuel Blum, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, 1973. https://doi.org/10.1016/S0022-0000(73)80033-9

[5] Joan Boyar, Lene M. Favrholdt, Shahin Kamali, and Kim S. Larsen. Online bin covering with advice. *Algorithmica*, 83(3):795–821, 2021. https://doi.org/10.1007/s00453-020-00728-0

[6] Joan Boyar, Shahin Kamali, Kim S. Larsen, and Alejandro López-Ortiz. Online bin packing with advice. *Algorithmica*, 74(1):507–527, 2016. https://doi.org/10.1007/s00453-014-9955-8

[7] Hans-Joachim Böckenhauer, Dennis Komm, Rastislav Královič, Richard Královič, and Tobias Mömke. On the advice complexity of online problems. *Proc.* 20th *ISAAC*, LNCS 5878, pages 331–340, 2009. https://doi.org/10.1007/978-3-642-10631-6_35

[8] Edward G. Coffman Jr., Joseph Y-T. Leung, and D.W. Ting. Bin packing: Maximizing the number of pieces packed. *Acta Informatica*, 9(3):263–271, 1978. https://doi.org/10.1007/BF00288885

[9] János Csirik and Vilmos Totik. Online algorithms for a dual version of bin packing. *Discrete Applied Mathematics*, 21(2):163–167, 1988. https://doi.org/10.1016/0166-218X(88)90052-2

[10] Michael R. Garey and David S. Johnson. *Computers and Intractability*, volume 174. Freeman San Francisco, 1979. ISBN:0-7167-1045-5

Study on Using Reinforcement Learning for the Monotone Boolean Reconstruction

Hasmik Sahakyan[1,3*], Gyula Katona[2] and Levon Aslanyan[1]
[1]Institute for Informatics and Automation Problems of the National Academy of Sciences, P.Sevak street 1, Yerevan 0014, Armenia
[2]Alfréd Rényi lnstitute of Mathematics, Budapest, Reáltanoda u.13-15 PF 127, 1364, Hungary
[3]French University in Armenia, D.Anhaght street 10/1, Yerevan 0037, Armenia
E-mail: hsahakyan@sci.am, katona.gyula.oh@renyi.hu, lasl@sci.am
[*]Corresponding author

Keywords: Monotone Boolean functions, reinforcement learning, combinatorial optimization

Received: April 14, 2023

This paper investigates the feasibility of using reinforcement learning to solve combinatorial optimization problems, in particular, the problem of query-based monotone Boolean function reconstruction. The monotone Boolean function reconstruction problem is a typical combinatorial problem that reconstructs the function unambiguously with a minimum number of queries about the value of the function at the defined points, based on the monotonicity of the function. The Shannon complexity of the problem is of the order of $2^n/\sqrt{n}$, and the solution algorithm relies on complex constructions, which also add complexity in the form of memory and time. Additionally, there are problems of partial reconstruction, e.g., in the mining of associative rules, which do not fit into the developed solution formats. This necessitates exploring heuristic domains to attract additional resources to solve the problem. To this end, all elements of reinforcement learning - environment, agent, policy, etc. - are designed, and both exact and approximate algorithms are given to perform the necessary structural data transformations, as well as to calculate the reward, the value, and other operational data of the algorithm. The focal point of the considerations is a subclass of monotone Boolean functions related to the well-known shadow minimization theorem of layer-by-layer characterized functions. Preliminary experiments have been started and they require follow-up intensive actions.

Povzetek: V raziskavi so avtorji preučili uporabo okrepljenega učenja za rekonstruiranje monotone Booleanove funkcije z minimalnim številom poizvedb. Predlagali so algoritme za točno in približno reševanje problema ter izvedli začetne simulacije.

1 Introduction

Many problems with monotone Boolean functions (MBFs) appear not only in logical and physical level design of systems, but also in artificial intelligence models, computation learning theory, hypergraph theory, and other areas. MBFs are used to encode extremely important constructions in various combinatorial optimization problems; they provide a natural way to describe satisfiable subsets of finite constraint collections. Extreme points of MBFs correspond to maximal compatible subsets of constraints, such as sets of linear inequalities, closed sets of frequent elements in association rule searches, etc. A number of applications (e.g., wireless sensor networks, dead-end tests of tables, data mining [2,3]) are based on MBF optimization, where MBFs are represented not in direct form, but by using chains and anti-chains [29]. Other similar applications can be added to this description [6,8,17].

There are a number of effective tools and methods for analyzing MBFs, and new approaches are constantly being sought, investigated, and applied. Well-known open problems include the reconstruction of bounded classes of Boolean functions with randomization of queries and functions, and the use of cube-splitting and chain-splitting of the Boolean domain [2].

A well-known problem concerning MBFs is the identification problem – the recognition of an unknown MBF of n variables by using membership queries. Hansel's algorithm [15], based on partitioning the binary cube into special non-intersecting chains, provides optimal reconstruction in the sense of Shannon complexity. In practical implementations, it is not necessary to build and store all chains in computer memory; according to [27,28,31], this can be done with simple procedures and an algebra that provides all the necessary answers to questions concerning the chains and queries mapped to these chains.

The problem of recognition of monotone Boolean functions with n variables, for $n = 2, \cdots, 4$ is investigated in [4], where the authors compare the complexity of different types of optimal (relative to the depth or the number of realizable nodes) decision trees with hypotheses.

PAC (probably approximately correct) type learning algorithms of MBFs are considered in a number of papers [25,32]. An algorithm that learns any monotone Boolean function f of n variables, to any constant accuracy, under the uniform distribution, in time polynomial in n and in the decision tree size of f is achieved in [32].

In order to obtain solutions and approximations using MBF classes, it is necessary to reconstruct the function itself from examples/values of points. One of the key tools here is the well-known Kruskal-Katona theorem [13,14], which describes the exact optimal monotone construction for a given set of parameters. In this way, KK-MBF class of MBFs is formed, being a special and attractive class for recognition.

In recent years, machine learning (ML) has been seen as an additional technological resource for solving combinatorial optimization problems (CO) [18,22,33]. Numerous machine learning techniques have been used to solve various combinatorial optimization problems, such as reinforcement learning to train a deep Q-net [18] which is applied to the Traveling Salesman Problem (TSP) [16,26], graph convolution networks [22], pointer networks and learning-to-prune framework [33], decision trees and neural networks to classify discrete images with different structural properties using projection data [12], and others.

In this paper, we investigate the possibility of using ML to train an MBF recognition heuristic that uses structures of Boolean function classes and stepwise recognition based on membership queries, as well as similarity and approximation studies between MBF classes and fragments. Among machine learning techniques, we distinguish and apply Reinforcement Learning (RL) – an approach similar to the query-based recognition. In order to apply RL to combinatorial optimization problems, the problem must be modelled as a sequential decision-making process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution. A fundamental question arises here, which requires a detailed analysis – whether it is possible to apply RL effectively for considered CO problem. Subjects for discussion include the compilation of acceptable sets of states and actions, possible and useful reward definitions, complexity aspects, and certainly, validation and estimation of approximations. These questions constitute the main topics of investigation of our present study. We propose an RL structure to solve the MBF recognition problem through a model (actions, states, and rewards) and an RL-MBF algorithm. Experiments involve simulations with a traditional RL algorithm SARSA, using instances of typical MBF classes.

The rest of the paper is organised as follows. Section 2 presents necessary definitions, preliminaries, and basic theoretical concepts of the RL and MBF recognition, respectively. Section 3 and 4 describe the proposed technique and model, RL structure and components for the MBF recognition, and RL MBF algorithm and data structures, respectively. Discussions and evaluation of the model are given in Section 5. The paper ends with some concluding remarks.

2 Preliminaries

2.1 Monotone Boolean function recognition

Let $B^n = \{(x_1, \cdots, x_n) \mid x_i \in \{0,1\}, i = 1, \cdots, n\}$ denote the set of vertices of the n-dimensional binary (unit) cube. Vertices of B^n are obtained by assigning values to the binary variables $x_1, \cdots, x_n$. Let $\alpha = (\alpha_1, \cdots, \alpha_n)$ and $\beta = (\beta_1, \cdots, \beta_n)$ be two vertices of B^n. Then, α precedes β (denoted as $\alpha \preccurlyeq \beta$) if and only if $\alpha_i \leq \beta_i$ for $1 \leq i \leq n$. If at the same time $\alpha \neq \beta$ then α strictly precedes β (denoted as $\alpha \prec \beta$). α and β are comparable if $\alpha \preccurlyeq \beta$ or $\beta \preccurlyeq \alpha$, otherwise, they are incomparable. A set of incomparable vertices in B^n is also called a Sperner family. A (growing) chain is a sequence of vertices such that the i-th vertex in the sequence is obtained from the $(i-1)$-th vertex by replacing a "0" component with "1".

The *Hamming distance* between α and β is the number of positions at which $\alpha_k \neq \beta_k$, $1 \leq k \leq n$.

For a given vertex α of B^n, we define two *intervals* of vertices as:

$$[\alpha, \tilde{1}] = \{\beta \in B^n | \alpha \preccurlyeq \beta \preccurlyeq \tilde{1}\}$$
$$[\tilde{0}, \alpha] = \{\beta \in B^n | \tilde{0} \preccurlyeq \beta \preccurlyeq \alpha\}$$

where $\tilde{1}$ and $\tilde{0}$ are vertices of B^n with all components being 1s and 0s, respectively.

Boolean function $f: B^n \to \{0,1\}$ is called *monotone* if for every two vertices $\alpha, \beta \in B^n$, if $\alpha \prec \beta$ then $f(\alpha) \leq f(\beta)$. Vertices of B^n, where f takes value "1" are called *units* or true points of the function; vertices, where f takes value "0" are called *zeros* or false points of the function. α^1 is a *lower unit* (or minimal true point) of the function if $f(\alpha^1) = 1$, and $f(\alpha) = 0$ for every $\alpha \in B^n$, such that $\alpha \prec \alpha^1$. α^0 is an *upper zero* (maximal false point) of the function if $f(\alpha^0) = 0$, and $f(\alpha) = 1$ for every $\alpha \in B^n$ such that $\alpha^0 \prec \alpha$. $\min T(f)$ and $\max F(f)$ denote the sets of minimal true points and maximal false points, respectively. Obviously, $\min T(f)$ and $\max F(f)$ are Sperner families in B^n.

Formally, the work with MBFs started in 1987, with the issue of counting their number [9]. The first algorithmic and complexity-related considerations belong to [19], where, in particular, the valuable concept of *resolving subsets* was introduced. The final asymptotic estimate about the number of MBFs of n variables was obtained in [20]. The technique on how to introduce and analyze MBFs, basically, is presented in [13,14,15,1,2,31,27,28,11].

The Hansel chain structure [15] was invented in 1966 and plays one of the central roles in MBF-related algorithmic techniques. The method is based on the partitioning of B^n into $\binom{n}{\lfloor n/2 \rfloor}$ pairwise non-intersecting chains (Hansel chains), arranged symmetrically about the middle layers of B^n, which have the following properties: 1) The number of chains of the length $n-2p+1$ is $\binom{n}{p} - \binom{n}{p-1}$ for $0 \leq p \leq \lfloor n/2 \rfloor$; 2) For any three elements $\alpha^{i_1} \prec \alpha^{i_2} \prec \alpha^{i_3}$ of a chain of length $n-2p+1$, their

relative complement β belongs to a chain of length $n - 2p - 1$.

The next valuable step to this was done by Tonoyan [31], who invented a set of simple procedures (chain algebra) that serve all the actual queries about Hansel chains, providing a technical solution to all the problems related to algorithms with Hansel chains, without constructing and keeping them in computer memory. In continuation, [27] presented a slightly modified and simplified version by using two tools: enumeration of all chains, and a procedure of finding the i-th vertex of the j-th chain. An optimised procedure is used to propagate newly found values to the chains, by a divide-and-conquer manner.

In the MBF recognition problem using membership queries, the goal is to determine an unknown MBF of n variables using as few queries as possible. The function can be fully recognized by finding all its upper zeros (and/or lower units) [19]. The Shannon complexity of finding all upper zeros (lower units) of an arbitrary monotone Boolean function of n variables is $\binom{n}{\lfloor n/2 \rfloor} + \binom{n}{\lfloor n/2 \rfloor + 1}$ [15].

General steps in any optimal in the sense of Shannon function algorithm of recognition of monotone Boolean function f are the following:

1. Partitioning B^n into Hansel chains;
2. Choosing a chain j (usually, chains are in increasing order of length) and a vertex α on j and querying $f(\alpha)$; On each chain, its "undetermined values" constitute an interval, and α is chosen as the middle point of the interval. Then the query response $f(\alpha)$ determines values in one half of the interval;
3. Propagating $f(\alpha)$ to other chains/vertices of B^n whenever it is possible using the monotonicity of f;
4. Repeating steps 2-3 until f is fully recognized (i.e., $f(\alpha)$ is obtained on all $\alpha \in B^n$).

Another recognition structure is used in [28]. For even n, B^n is split according to two variables and the recognition in every sub-cube starts from its two middle layers. For odd n, firstly B^n is split according to one variable, then as each sub-cube now has even size, the procedure for even sizes is applied. This provides optimal recognition of all MBFs in the sense of Shannon complexity. Unfortunately, while simple and attractive, this approach cannot be used in practical algorithms for arbitrary functions. Finally, it is worth mentioning the work [11] that considers not the Shannon complexity but the individual complexity of MBF given by its resolving set size. The core consideration is that for a given vertex α, if $f(\alpha) = 0$, an upper-zero of f can be found in no more than n queries. The same holds with a lower-one when $f(\alpha) = 1$. This gives a recognition complexity bounded by n times the resolving set size of f.

In general, tasks related to the recognition of MBFs may have different formulations. One task is to recognize a particular unknown function, knowing that it belongs to the class of MBFs or to one of its subclasses. Another task is to start with partial knowledge about the unknown function. One more case is when the number of queries is restricted by some number K and the goal is to maximize the recognized part of the function.

Similar problems can be formulated for specific classes of Boolean functions. Examples of classes are as follows:

KK-MBF Kruskal-Katona MBF arises as a result of the shadow minimization theorem [21,13,14]. KK-MBF intersects the layers of the cube along their initial segments of lexicographic order. The complement of the KK-MBF area in B^n has a similar property; it is related to the initial segments of co-lexicographic order. Partial KK-MBF is a partial MBF, where 0s of the function form initial segments of co-lexicographic order, and 1s form initial segments of lexicographic order on layers.

Symmetric MBF This is a trivial class of functions that take constant value on the cube layers. Examples are majority functions, parity functions, and others. Partial symmetric MBF is when the function is not determined on some layers of the cube.

Threshold MBF Functions are defined by a linear inequality of weighted sums of variables.

Matroid MBF Monotone Boolean function f is called a matroid function if for each $\alpha, \beta \in \min T(f)$ with $\alpha_i = 1, \beta_i = 0$, there exists a coordinate j with $\alpha_j = 0$, $\beta_j = 1$ such that vertex α', obtained from α by replacing α_i with 0 and α_j with 1, belongs to $\min T(f)$.

The combinatorial complexity of reconstruction in subclasses is not well studied. It is known that symmetric functions, with zeros and ones separated by two middle layers of the cube, are the most difficult functions for query-based reconstruction when only the monotonicity of the function is given. If it is known that the function belongs to the class of symmetric functions, the reconstruction can be done by $logn$ queries. The same function also belongs to $KK - MBF$, and the combinatorial complexity in this case does not exceed $nlogn$.

Other known and investigated classes of important MBFs, such as 2-monotonic positive functions k-tight functions, can be found in [7,24].

2.2 Reinforcement learning

Reinforcement learning (RL) [30,23] is an algorithmic implementation of the natural learning process in which there is a systematic and analytical interaction with the environment. The basic elements of RL are the environment and the learner agent that interacts with it. When interacting with the environment, the agent performs actions, and the environment responds through a reward. Accordingly, RL algorithms distinguish the

following main elements: *policy*, which is a mapping of environmental situations/states to actions and is the core of the reinforcement learning agent; *reward*, which characterizes the immediate reaction of the environment to the action; the *value* of state s or the value of state-action pair (s, a). The value function $V(s)$ is the expectation of future rewards when starting from state s and following some policy π; and the action-value function $Q(s, a)$, which is the expectation of future rewards when starting from state s, taking action a and following some policy π.

RL algorithms can be divided into policy-based and value-based methods. In the case of policy-based methods, the policy is approximated directly, whereas value-based methods focus on approximating a value function, which is a measure of policy quality for some state-action pair in the given environment. Value-based methods first compute the action-value function $Q(s, a)$ as the expected reward of the policy, given state s and taking action a. The agent's policy then corresponds to the choice of action that maximizes $Q(s, a)$ for that state. The main difference between the value-based approaches is how to estimate $Q(s, a)$ accurately and efficiently.

Another important element of RL algorithms is the *model* of the environment, which may be unknown or partially known. The nature of RL elements - deterministic or stochastic, representation by function, table, or table being filled, presence of an environment model, nature of the applied policy, etc. – defines, in many respects, its mathematical model and problems of implementation.

RL is based on the concept of Markov decision-making processes (MDP) [5]. MDPs are structured from finite sets of actions, states, policies and a state transition model. A learner agent interacts with the environment in a sequence of steps in time (t): (i) the agent receives a representation of the current environment (some state); (ii) chooses and performs an action according to its policy; (iii) receives a reinforcement/reward; and (iv) enters a new state of the environment. The goal in MDP and reinforcement learning is to learn a policy π that maximizes the expected sum of future rewards.

The epsilon-soft (ε-soft) is an example of action selection policy in RL, where the probability of all actions given a state s is greater than some minimum value. The epsilon-greedy (ε-greedy) policy is a specific instance of an ε-soft, where the policy is applied according to the following equation:

$$\pi(s) = \begin{cases} a^* \; with \; probability \; 1 - \epsilon \\ a_a \; with \; probability \; \epsilon \end{cases}$$

where $\pi(s)$ is the decision policy for the current state s, a^* is the best estimated action for the state s at the current time, and a_a is a random action selected with probability ϵ [30].

In practice it is rare that an agent learns optimal policy; only an approximation can be achieved in various learning algorithms. The available memory is an important constraint; large amount of memory is often required to build up approximations of value functions, policies, and models. In tasks with small, finite state sets, it is possible to form these approximations using learning tables with one entry for each state (or state–action pair). In many cases of practical interest, however, there are far more states than could possibly be entries in a table. In these cases the functions must be approximated.

The online nature of reinforcement learning allows optimal policies to be approximated in ways that prioritize learning to make good decisions for frequently encountered states, at the expense of less effort for infrequently encountered states. This is one key property that distinguishes reinforcement learning from other approaches to approximately solving MDPs.

SARSA, a counterpart of Q-learning approach [30], is one of the common approximate value-based methods. It is based on temporal difference learning, where update estimates are partially based on other learned estimates, without waiting for a final outcome. SARSA iteratively makes the following update of the action-value on transition to the next state:

$$Q_{t+1} \coloneqq Q_t(s, a) + \alpha[r(s, a) + \gamma Q_t(s', a') - Q_t(s, a)]$$

where s is the state and a is the action at the current step t, respectively; s' is the state and a' is the action at the next step $t + 1$. $Q_t(s, a)$ is the action-value at time t for the state-action pair (s, a). Q_{t+1} is the update at time $t + 1$ by performing action a in state s; $r(s, a)$ is the reward received for the pair (s, a); α and γ are the parameters, where γ is the scalar discount factor, $0 < \gamma \leq 1$, that describes the agent preference between current and future reward; α is the learning rate, which controls overlap speed of new information.

In general, agent–environment interaction is naturally broken down into a sequence of separate episodes, and each action affects only the finite number of rewards subsequently received during the episode. The time steps of each episode are numbered starting anew from zero.

SARSA Algorithm

Algorithm parameters: step size $\alpha \in (0,1]$, small $\epsilon > 0$;
Initialize $Q(s, a)$ arbitrary for each pair (s, a); except that $Q(terminal\ state, a)$ must be 0 for all a;
Loop for each episode
  Observe state s;
  Select a using policy derived from Q, e.g., ϵ –greedy;
  Repeat
    Take action a;
    Receive reward $R(s, a)$;
    Observe the next state s';
    Select the new action a' using policy derived from Q, e.g., ϵ –greedy;
    Update $Q(s, a)$ with
    $Q(s, a) \coloneqq$
    $Q(s, a) + \alpha[R(s, a) + \gamma Q(s', a') - Q(s, a)]$;
    $s \coloneqq s', a \coloneqq a'$
  Until s is terminal

3 RL components for MBF recognition

The idea of using RL for MBF recognition is to replace the difficult task of determining the next vertex of the

binary cube B^n in combinatorial algorithms (to query the function value) with using examples, parameters learning and mimicking the best behaviour. The whole RL framework is composed of combinatorial preparations and learning heuristics. To understand the role of combinatorial and heuristic procedures in the overall RL structure, consider the data structures of the SARSA algorithm given in Figure 1.

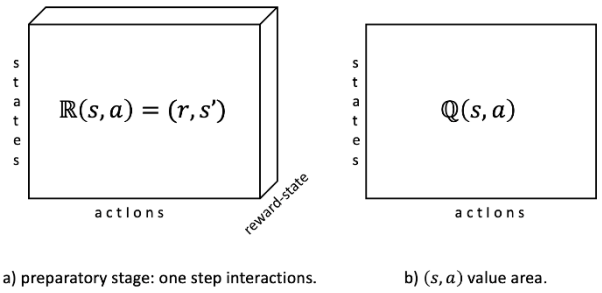
Data in part a) of **Error! Reference source not found.** is algorithmic in nature. We call this the Reward table. In the preparatory stage of the problem, it is necessary to define adequate reward values for all state-action pairs (s, a) (assuming that we have the state and action sets). The reward values will not be changed during the whole performance. The new state s' is also definite, in the sense that being in state s and performing action a will lead to s', defined by deterministic or stochastic transition rules. In this part, we will intensively use technologies largely implemented in algorithmic studies of MBF recognition. The table $\mathbb{R}$ can be completely or partially filled, containing only the values that have already been calculated by that moment.

Often, also in MBF recognition, the task of filling $\mathbb{R}$ is solved by searching or using other discrete, combinatorial considerations.

Data in part b) is in the operative domain controlled by the learning process. Initially, this data is initialized with zero values and then learned step-by-step. The learning is based on the learning method (given in [30] for SARSA), which is also based on the learning policy.

On the one hand, policy uses the table $\mathbb{Q}$, the part of it that is already updated, and on the other hand, it is extended with a "planning" step that assigns some arbitrary actions randomly. This is a kind of balance between known and unknown rewards. SARSA, having the current state s, selects action a using the policy derived from $\mathbb{Q}$, obtains (r, s') from the reward table, and selects the next action a' by the policy, and on this basis learns and updates the new value of $\mathbb{Q}$.

We conclude: the learning part of SARSA is universal; it works with the policy function π and the value table $\mathbb{Q}$, and these areas as well as the learning procedure are independent of the subject domain under consideration. The complementary part related to the reward table $\mathbb{R}$ defines one-step-rewards, which in our case is a combinatorial computational problem. Then, the concept of learning is to integrate these one-step-rewards into longer chains of predictions, thus intending to achieve the learning goal. For needs of learning over MBF classes, it is now clear that we will apply the combinatorial technique developed in this area.



a) preparatory stage: one step interactions. b) (s, a) value area.

Figure 1: Basic data structures used in typical SARSA

3.1 MDP model on MBF reconstruction

In a typical MBF recognition problem using membership queries, the goal is to determine an unknown MBF f of n variables by as few queries as possible. We can also consider the problem of determining whether an unknown Boolean function belongs to the class of monotone functions. Another case is when the number of queries is limited to some number K and the goal is to maximize the recognized part of the function.

In the following, we will point out a number of individual problems of this type that may arise in RL MBF applications, and their mathematical formulations. Suppose that the vertices of B^n are enumerated. Let x_j denote binary variables such that x_j takes value 1 if the j-th vertex of B^n is chosen to ask the value of the function, and 0 otherwise. c_j expresses the informativeness of the j-th vertex, i.e., the number of new evaluations of the function obtained by propagating the value of the j-th vertex.

The formalism of these steps, depending on the particular objectives, may be given as follows:

1) $min \sum_j x_j$, subject to $\sum_j c_j = 2^n$, this is the case when the function is to be entirely recovered and the goal is to minimize the number of queries;
2) $min \sum_j x_j$, subject to $\sum_j c_j \geq N$, this is the case when a certain part of the function is to be recovered by the minimum number of queries;
3) $max \sum_j c_j$, subject to $\sum_j x_j = K$, this is the case when the number of queries is restricted and the goal is to maximize the recognized part of the function.

Note that the values of c_j become known only after the vertices are selected. But regardless of the order of selection, $\sum_j c_j$ will have the same value at the end.

In order to apply RL to MBF recognition we need to reformulate it in terms of MDP, i.e., define the environment, states, actions, rewards, transition, and policy.

3.1.1 Environment

The basis of the RL environment for the MBF reconstruction considered in this paper, is the n-dimensional binary cube, its vertex set B^n, and the related sub-cube structure. The environment may need additional tools /known concepts/ such as Sperner systems, Hansel chains, Hamming distance, lexicographic order, and others.

3.1.2 States

The states represent some /partial/ knowledge about the solution, such us nested parts of the looked-for MBF, which we call *fragments*. An initial fragment can be the empty set, or we can start with some non-empty fragment. Thus, formally, each state s is a partially defined monotone Boolean function f^s, given through a pair (T_{f^s}, F_{f^s}) of some sets of true and false vertices.

An agent must travel through the remaining (undetermined) vertices of the cube and decide (according to the policy) which vertex to choose as a query. Thus, actions are vertices chosen to ask the value of the function.

3.1.3 Rewards

Since the agent's instrument of interaction with the environment is a query $\alpha \in B^n$, the response of the environment should characterize how successful the query is, having a predetermined partial monotone Boolean function available when the query is performed. Thus, the reward will express the "informativeness" of the chosen vertex-action α in the sense of spreading the value $f(\alpha)$ to other vertices due to the monotonicity of f.

3.1.4 Transition/update

Depending on the chosen vertex and the function value, new state s' will be determined, which is a new partially defined function $f^{s'}$ given through a new pair $(T_{f^{s'}}, F_{f^{s'}})$, such that $T_{f^s} \subseteq T_{f^{s'}}$ and $F_{f^s} \subseteq F_{f^{s'}}$.

3.1.5 Terminal state

The process will be continued until the termination criterion is reached, which according to the defined objectives is as follows:

1) f is completely recovered,
2) f is recovered by a certain percentage,
3) a certain number of steps are done.

3.1.6 Policy

Environment models, when known, are used for planning, by which we mean an arbitrary approach of deciding to apply actions by considering possible future situations before they actually occur. Methods for solving reinforcement learning problems that use models and planning are called model-based methods, in contrast to methods without models that learn by trial and error.

The goal of all reinforcement learning algorithms is to find a policy (the way of choosing actions), which would consistently allow the agent to gain a lot of rewards. For MBF recognition, the goal is to find a policy to get as many known values of the function as possible (ideally, this will be the whole function).

Initially, we will start with the following: being in any state s, actions are chosen from the area $B^n \setminus (T_{f^s} \cup F_{f^s})$, using ε-greedy method. According to MBFs structure, vertices of $\bigcup_{\alpha \in T_{f^s}} [\alpha, \tilde{1}]$ are true vertices, and vertices of $\bigcup_{\alpha \in F_{f^s}} [\tilde{0}, \alpha]$ are false vertices of f. It is then natural to exclude this part from further consideration. Thus, another policy may be: being in any state s, choose actions from the area $B^n \setminus (\bigcup_{\alpha \in T_{f^s}} [\alpha, \tilde{1}] \cup \bigcup_{\alpha \in F_{f^s}} [\tilde{0}, \alpha])$ using ε-greedy method.

Moreover, special policies may be defined that exploit structures of MBF, and are based on Hamming distances between selected points in T_{f^s} and F_{f^s}.

3.2 Initial fragments, policies, updates, and rewards

We consider three types of initial fragments.

3.2.1 Partially defined monotone Boolean functions given by Sperner families

The input fragment (corresponding to the initial state s) is given through a pair of feasible/compatible Sperner families (T_{f^s}, F_{f^s}), one for the lower units, and one for the upper zeros of a partially defined monotone Boolean function f^s, as shown in Figure 2 a).

Policy 1 - actions are selected in $B^n \setminus (T_{f^s} \cup F_{f^s})$ using ε-greedy method.

Actions

Action a in a given state s is a vertex α in $B^n \setminus (T_{f^s} \cup F_{f^s})$, chosen to ask the value $f(\alpha)$.

New states

Depending on $f(\alpha)$ the following operations are to be performed:

For $f(\alpha) = 1$

i. if α is not comparable with any vertex of T_{f^s}, then α is added to T_{f^s}, thus the new state s' is defined by the pair $(T_{f^{s'}}, F_{f^{s'}})$, where
$$T_{f^{s'}} = T_{f^s} \cup \{\alpha\} \text{ and } F_{f^{s'}} = F_{f^s}$$
ii. if α precedes some vertex/vertices β of T_{f^s}, then α is added to T_{f^s} and all vertices greater than α are removed from T_{f^s}, thus
$$T_{f^{s'}} = (T_{f^s} \cup \{\alpha\}) \setminus \{\beta | \beta \in T_{f^s} \text{ and } \beta \succ \alpha\}$$
iii. if there is a vertex β of T_{f^s} that precedes α, then the partial solution /state/ is not changed,
$$T_{f^{s'}} = T_{f^s} \text{ and } F_{f^{s'}} = F_{f^s}$$

For $f(\alpha) = 0$

iv. if α is not comparable with any vertex of F_f, then α is added to $F_{f^{s'}}$, thus
$$F_{f^{s'}} = F_{f^s} \cup \{\alpha\}, T_{f^{s'}} = T_{f^s}$$

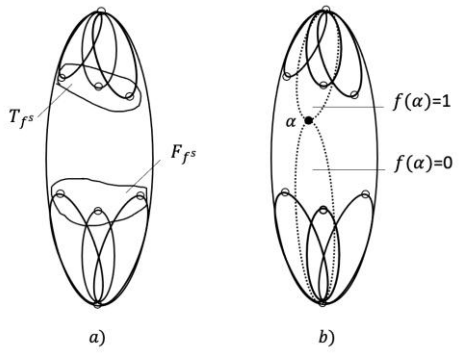


Figure 2: Partial MBF given by Sperner systems T_{f^s} and F_{f^s}

v. if there is a vertex/vertices β of F_{f^s} that precedes α, then α is added to F_{f^s} and all vertices that precede α are removed from F_{f^s}, thus $F_{f^{s\prime}} = F_{f^s} \cup \{\alpha\} \backslash \{\beta | \beta \in F_{f^s} \; and \; \beta \prec \alpha\}$

vi. if f α precedes some vertex β of F_{f^s}, then the partial solution /state/ is not changed, $T_{f^{s\prime}} = T_{f^s}$ and $F_{f^{s\prime}} = F_{f^s}$.

An illustration is given in Figure 2 b).

Rewards

We will use approximate reward calculations based on Hamming distances between the vertex α and the vertices of T_{f^s} and F_{f^s}.

Reward 1

The reward is 0 for the cases iii and vi, otherwise, it is the average Hamming distance, when averaged over all distances from α to the vertices of T_{f^s} and F_{f^s}:

$$R = \frac{\sum_{\gamma \in (T_{f^s} \cup F_{f^s})} \rho(\alpha, \gamma)}{\left|T_{f^s} \cup F_{f^s}\right|}$$

Reward 2

The reward is 0 for the cases iii and vi, otherwise, it is the average Hamming distance, when the average is over the two farthest distances from α in T_{f^s} and F_{f^s}:

$$R = \frac{\max\limits_{\gamma \in T_{f^s}} \rho(\alpha, \gamma) + \max\limits_{\gamma \in F_{f^s}} \rho(\alpha, \gamma)}{2}$$

3.2.2 Partially defined monotone Boolean functions given by bunch of intervals

The input fragment (corresponding to the initial state s) is given through a pair of bunches of intervals $(\cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}], \cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])$; initially, T_{f^s}, F_{f^s} are feasible/compatible Sperner families, one for the lower units, and one for the upper zeros of a partially defined monotone Boolean function f^s.

Policy 2 - actions are selected in $B^n \backslash (\cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}] \cup \cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])$ using ε-greedy method.

We have two options:

1. Keep all intervals $\cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}]$ and $\cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])$, - this requires extra memory;
2. Exclude all vertices greater than any vertex of T_{f^s}, and lower than any vertex of F_{f^s} – this requires extra computations.

Actions

Action a in given state s is a vertex α in $B^n \backslash (\cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}] \cup \cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])$, chosen to ask the value $f(\alpha)$.

New states

Depending on $f(\alpha)$ the following operations are to be performed ($T_{f^{s\prime}}$ and/or $F_{f^{s\prime}}$ may not be Sperner families):

i. if $f(\alpha) = 1$ then $T_{f^{s\prime}} := T_{f^s} \cup \{\alpha\}$ and $F_{f^{s\prime}} := F_{f^s}$,

ii. if $f(\alpha) = 0$ then $F_{f^{s\prime}} := F_{f^s} \cup \{\alpha\}$ and $T_{f^{s\prime}} := T_{f^s}$,

the new state is $(\cup_{\alpha \in T_{f^{s\prime}}}[\alpha, \tilde{1}], \cup_{\alpha \in F_{f^{s\prime}}}[\tilde{0}, \alpha])$.

Rewards

Reward 1

$$R = \left|(\cup_{\alpha \in T_{f^{s\prime}}}[\alpha, \tilde{1}] \backslash \cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}]) \cup (\cup_{\alpha \in F_{f^{s\prime}}}[\tilde{0}, \alpha]) \backslash (\cup \cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])\right|$$

Reward 2

$$R = \frac{\left|(\cup_{\alpha \in T_{f^{s\prime}}}[\alpha, \tilde{1}] \backslash \cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}]) \cup (\cup_{\alpha \in F_{f^{s\prime}}}[\tilde{0}, \alpha]) \backslash (\cup \cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])\right|}{\left|B^n \backslash (\cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}] \cup \cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])\right|}$$

$$= \frac{\left|(\cup_{\alpha \in T_{f^{s\prime}}}[\alpha, \tilde{1}] \backslash \cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}])\right| + \left|(\cup_{\alpha \in F_{f^{s\prime}}}[\tilde{0}, \alpha]) \backslash (\cup \cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha])\right|}{2^n - \left|\left(\cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}]\right)\right| - \left|\cup_{\alpha \in F_{f^s}}[\tilde{0}, \alpha]\right|}$$

Reward 3

For obtaining an approximate value for the reward, we can apply Bonferroni inequalities [35,36] for approximating the value of corresponding inclusion-exclusion formula. For example, $R = \left|[\alpha', \tilde{1}] \backslash \cup_{\alpha \in T_{f^s}}[\alpha, \tilde{1}]\right|$ can be approximated in the following way.

We suppose that $h_1, h_2, \ldots, h_q$ are the smallest vertices of T_{f^s} that are greater than α'.

Denote $H_p = [h_p, \tilde{1}]$ for $p = 1, \cdots, q$. The exact reward value can be expressed by the inclusion-exclusion formula:

$$R = 2^{|\alpha\prime|} - \left|\cup_{p=1}^{q} H_p\right| = 2^{|\alpha\prime|} - \sum_p |H_p|$$

$$+\sum_{i<j}|H_i \cap H_j| - \sum_{i<j<k}|H_i \cap H_j \cap H_k| + \cdots (-1)^q |H_1 \cap H_2 \cap \ldots \cap H_q|,$$

where $|\alpha'|$ denotes the size of the interval $[\alpha', \tilde{1}]$, and $2^{|\alpha'|}$ is the number of vertices in $[\alpha', \tilde{1}]$.

The sum of some first terms in the formula is alternately an upper bound and a lower bound, which is the key for approximations.

3.2.3 Points on chains

The input fragment is composed as follows: split B^n into Hansel chains (Figure illustrates the Hansel chains in B^5), and compose a list $L = (l_1, l_2, \cdots, l_{\binom{n}{\lfloor n/2 \rfloor}})$ of their lengths; for example, $L = (6,4,4,4,4,2,2,2,2,2)$ for B^5. Take at random two numbers in $[1, l_j]$ for each chain j. The smaller number will indicate the initial maximal false point, and the larger number will indicate the initial minimal true point of the chain.

Compose T_{f^s} and F_{f^s} from the initial minimal true and maximal false points of the chains, respectively. Let (α_j, β_j), $\alpha_j \prec \beta_j$, be the current maximal false and minimal true vertices of the j-th chain. Compose intervals $[\alpha_j, \beta_j]$ on the chains and call them "uncertainty intervals" of the chains.

Policy 3 - actions are chosen in $\bigcup_j [\alpha_j, \beta_j]$ using ε-greedy method.

Actions

Action a in given state s is a vertex γ in $\bigcup_j [\alpha_j, \beta_j]$ chosen to query the value $f(\gamma)$.

New states

Let γ belongs to $[\alpha_j, \beta_j]$

i. if $f(\gamma) = 1$ then γ replaces the true point α_j of the chain, $\alpha'_j := \gamma$, resulting a new smaller uncertainty interval $[\alpha'_j, \beta'_j]$, $\beta'_j = \beta_j$. Moreover, $f(\gamma)$ can be spread to the vertices of the uncertainty intervals of all chains, thus replacing their true points (for simplicity we may omit this);
ii. if $f(\gamma) = 0$ then γ replaces the false point β_j of the chain, $\beta'_j := \gamma$, resulting a new smaller uncertainty interval $[\alpha'_j, \beta'_j]$, $\alpha'_j = \alpha_j$. Moreover, $f(\gamma)$ can be spread to the vertices of the uncertainty intervals of all chains, thus replacing their false points (for simplicity we may omit this).

We note that any optimal MBF recognition algorithm that works with Hansel chains chooses the middle points of the intervals to ask the value of the function. But the goal here is to determine the most informative point for a given function depending on its structure.

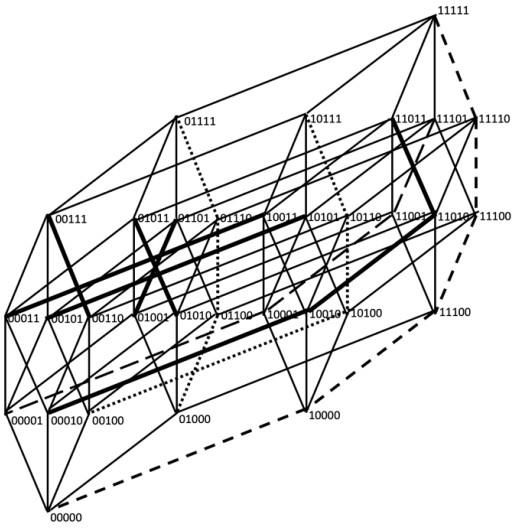


Figure 3: Hansel chains in B^5

Rewards

The reward for action α in state s depends on the "informativeness" of the vertex.

Reward 1

The reward is the difference between the summary lengths of the new and previous uncertainty intervals:

$R = \left| \bigcup_j [\alpha_j, \beta_j] \setminus \bigcup_j [\alpha'_j, \beta'_j] \right|$

Policy 4 - actions are chosen starting from the largest intervals using ε-greedy method.

Actions

An action a in a given state s is a vertex γ in the largest $[\alpha_j, \beta_j]$ chosen to ask the value $f(\gamma)$.

New states and rewards can be defined as for the previous case.

4 RL MBF algorithm

In this section we describe an RL algorithm for the case when states are defined by Sperner families, and the objective is to maximize the recognized part of the function, when the number of queries is restricted by some natural number K,

$max \sum_j c_j$
subject to $\sum_j x_j = K$.

The proposed algorithm is based on SARSA method. The policy π and the reward function R can be any of those defined in the previous section.

RL-MBF algorithm is composed of several parts:

Set parameters α, γ, policy π, reward function R, number of episodes N (episodes will be explained in Section 4.2)
Initialize RL-MBF variables and constants, including the cube size n, and integers $k, l, 0 \le k, l \le n$ to characterise Sperner families

Generate initial state s (consists of two Sperner families, one is for k minimal true points, and another is for l maximal false points)
Get feasibility area (this is "feasible" part of B^n for selecting valid query vertices, thus, excluded initial two Sperner families, or Sperner families along with the corresponding intervals, according to the policy π)
Create root vertex in Q-graph according to the initial state s;
Repeat for each episode
  Select vertex α /action a/ in the feasibility area according to the policy π
  Repeat
    Query the value $f(\alpha)$
    Get new state s' (according to the value $f(\alpha)$ and the policy π)
    Calculate Reward for (s, a) (according to Reward function R)
    Update feasibility area (remove the "difference" according to the policy π)
    Select new vertex α' /action a'/ in the feasibility area according to the policy π
    Update Q-table (update Q-graph using s, a, s', a' according to SARSA)
    Calculate cumulative reward
    $s := s'$
    $a := a'$
  until K steps are done
until N episodes are done

4.1 Functions used in RL-MBF

In this section we describe some of the functions used in RL-MBF algorithm.

Let $\alpha = (a_1, \cdots, a_n)$ and $\beta = (b_1, \cdots, b_n)$ be vertices of B^n. α precedes lexicographically β if either there exists an integer k, $1 \le k \le n$, such that $a_k < b_k$ and $a_i = b_i$ for $i < k$, or $\alpha = \beta$. The vectors of B^n are in a lexicographic order in the sequence $\alpha_0, \alpha_1, \cdots, \alpha_{2^n-1}$ if α_i precedes lexicographically α_j, for $0 \le i < j \le 2^n - 1$.

The *serial number* of the vertex $\alpha = (a_1, \cdots, a_n)$ is the natural number $a_1 2^{n-1} + a_2 2^{n-2} + \cdots + a_n 2^0$, whose binary representation is $a_1 a_2 \cdots a_n$. When the vectors of B^n are in a lexicographic order, their serial numbers form the sequence $0,1, \cdots, 2^n - 1$.

Generate initial state s and Get feasibility area (for Policy 1)

$T_{f^s} := \emptyset$; $F_{f^s} := \emptyset$; $F = \{0,1, \cdots, 2^n - 1\}$; k, l;
$t := 0$;
repeat
  r:= random number in F;
  find vertex α_r, with the serial number r;
  if α_r is not comparable with any vertex of T_{f^s} then
    $T_{f^s} := T_{f^s} \cup \{\alpha_r\}$;
    $F := F \backslash \{r\}$;
    $t = t + 1$;
until $t = k$
$t := 0$;
repeat
  r:= random number in F;
  find vertex α_r with the serial number r;
  if α_r is not comparable with any vertex of F_{f^s} and is not greater of any vertex of T_{f^s} then
    $F_{f^s} := F_{f^s} \cup \{\alpha_r\}$;
    $F := F \backslash \{r\}$;
    $t = t + 1$;
until $t = l$
$s := T_{f^s} \cup F_{f^s}$;

Generate initial state s and Get feasibility area (for Policy 2)

$\hat{T}_{f^s} := \emptyset$; $\check{F}_{f^s} := \emptyset$; $F = \{0, \cdots, 2^n - 1\}$; k, l;
repeat
  r= random number in F;
  find vertex α_r with the serial number r;
  $\hat{T}_{f^s} := \hat{T}_{f^s} \cup [\alpha_r, \tilde{1}]$;
  $F := F \backslash \{serial\ numbers\ of [\alpha_r, \tilde{1}]\}$;
until k steps are done

repeat
  r:= random number in F;
  find vertex α_r with the serial number r;
  $\check{F}_{f^s} := \check{F}_{f^s} \cup [\tilde{0}, \alpha_r]$;
  $F := F \backslash \{serial\ numbers\ of\ [\tilde{0}, \alpha_r]\}$;
until l steps are done
$s := \hat{T}_{f^s} \cup \check{F}_{f^s}$;

Get new state s' and Update feasibility area (for Policy 2)

If $f(\alpha) = 1$
  then
    $\hat{T}_{f^{s'}} := \hat{T}_{f^s} \cup [\alpha, \tilde{1}]$;
    temp:= $F \backslash \{serial\ numbers\ of\ [\alpha, \tilde{1}]\}$;
    diff:= F\temp; F :=temp;
  else
    $\check{F}_{f^{s'}} := \check{F}_{f^s} \cup [\tilde{0}, \alpha]$;
    temp:= $F \backslash \{serial\ numbers\ of\ [\tilde{0}, \alpha]\}$;
    diff:= F\temp;
    F :=temp;

RL-MBF algorithm, as well as functions used can be described in a similar way for the case, where states are defined by points on chains.

4.2 Data structure for Q-table

As mentioned earlier, agent–environment interaction is broken down into separate episodes, and each action affects only the finite number of rewards subsequently received during the episode.

In the case of MBF recognition, each episode starts with the same initial fragment (e.g. two initial Sperner families along with the corresponding intervals) and has fixed length K. Theoretically, an action can be any vertex of B^n, excluding the vertices of the initial fragment, and a

state can be any partial monotone Boolean function containing the initial fragment.

Within a single episode, a sequence of steps leads to a sequence of states that correspond to a series of nested partial monotone Boolean functions. Thus, "feasibility" area for choosing actions becomes more restricted from state to state. In terms of the reward table R, states are assigned to the rows of the table, and actions are assigned to the columns. Each state has its area of "feasible" actions (a subset of columns). At each step of an episode, being in a state s and performing an action a, if a is a new action for the state s, then cell (s, a) is filled with the computed reward value $R(s, a)$, otherwise, the cell is already filled with the reward value. For the first episode, R is initially empty, and at each step exactly one cell is filled.

Regarding the state-action table Q, at each step of an episode, being in state s and performing action a, if a is a new action for the state s, then cell (s, a) is assigned 0, otherwise $Q(s, a)$ is updated according to the method used (e.g., SARSA). For the first episode, the table is initially empty.

Given the features of MBF, i.e. the nested structure of states and the restriction of the range of feasible actions from state to state, instead of storing Reward- and Value-tables we create an oriented weighted graph Q-graph as follows.

States are assigned to vertices and state-action pairs are assigned to edges. One vertex is mentioned as the root vertex that corresponds to the initial state; the root has no incoming edges.

Each episode is an oriented path in Q-graph, starting from the root and having length K. Each edge (s, a) is assigned two weights; one is the reward $R(s, a)$, the other is the state-action value $Q(s, a)$. For the first episode, the entire path is newly created; $R(s, a)$ are computed according to the reward function, and $Q(s, a)$ are assigned 0, then $Q(s, a)$ are updated according to the SARSA formula.

Obviously, paths for different episodes can overlap (i.e., the same state can be reached by different sequences of actions), possibly, making cycles.

In the current episode, being in some state s and choosing action a, we search among the outgoing edges from s; if edge (s, a) already exists, then the pair has already met, and $R(s, a)$ and $Q(s, a)$ have been calculated.

If edge (s, a) does not exist, we first look for vertex s', if it exists we add edge (s, a) from s to s', if it does not exist then we create new vertex s' and add edge from s to s'. Then we calculate $R(s, a)$ according to the reward function, assign $Q(s, a)$ to 0, and then update it according to SARSA formula.

In order to organize the search process for vertex s and edges originating from s, as well as Reward and Q tables, we introduce the following structure.

We create an array "Vertex_list", each element v of which has its name "v.name" and its set of outgoing edges "v.edge_list". "v.name" keeps states, and "v.edge_list" keeps all outgoing edges from v. Initially, "Vertex_list" consists of the root vertex v_0; "v_0.name" is the initial state, "v_0.edge.list" is empty.

Each element "edge: of "v.edge_list", in its turn, has its own name "edge.name", which is the chosen vertex-action, its reward "edge.reward", and its Q-value "edge.value".

Update is organized in the following way.

Update Q(s, a, s', a')

If a does not exist in "s.edge_list", and s' does not exist in "Vertex_list", then create a new graph vertex s' in "Vertex_list" with "s'.name"= s', and add outgoing edge a from s to s' by adding (s, a) into the "s.edge_list" of s; with "edge.value" 0; and "edge.reward" - calculated according to the reward function.

If a does not exist in "s.edge_list", but s' exists in "Vertex_list", then add new edge a from s to s' by adding s' into the "s.edge_list"; calculate "edge.reward" according to the reward function. Update "edge.value" according to:

$Q(s, a) \coloneqq Q(s, a) + \alpha[R(s, a) + \gamma Q(s', a') - Q(s, a)]$.

5 Discussions and evaluation of the model

The RL modeling of the MBF recognition problem passes through the stages of defining all the elements of RL for the MBF domain, as well as the algorithmic implementation of the computational processes of these components. Exact models based on Sperner families and Hansel chains are used, as well as approximate versions of both concepts and their computations are proposed. Hansel models require at least twice the amount of regular memory and reduce the spatial analysis of structures to chain analysis using simple division and the algebra of Hansel chains. The approximations in the calculations associated with Sperner systems turn to inclusion-exclusion type formulas when computing the cardinality of a bundle of intervals, where the sums of the initial sum segments provide lower and upper cardinality estimates in the required precision.

Coarser heuristics are also possible, due to the need to reduce the resulting computational complexity. The simplest heuristics address different kinds of query distances or propagation of points from vertices to current Sperner systems. The whole technique of this block proposes replacing the ideal environment with another, less intelligent and less precise one, with the hope that the necessary computation is simplified and that the RL process converges in a reasonable time frame.

The preliminary experiments conducted involved functions of small dimensionality and mainly focused on clarifying the possibilities of approaches with Sperner and Hansel systems. However, a sufficient volume of experiments has not yet been conducted, and the available volume of trials does not allow to obtain reasonable conclusions to be drawn about the effectiveness of one approach or the other. At this stage, a *Proof of Concept* type result has been established, and an experimental computing environment has been deployed for further work on the basis of the Armenian-French supercomputer

of the Institute for Informatics and Automation Problems of the National Academy of Sciences of Armenia.

6 Conclusion and future work

This paper investigated the feasibility of using reinforcement learning to solve combinatorial optimization problems, particularly the query-based reconstruction of monotone Boolean functions.

Acknowledgments

This paper is partially supported by grant №21T-1B314 of the Science Committee of MESCS RA.

References

[1] L. Aslanyan, "The discrete isoperimetry problem and related extremal problems for discrete spaces", Problemy kibernetiki, vol.36, pp. 85–128, 1979.

[2] L. Aslanyan and H. Sahakyan, "The splitting technique in monotone recognition", Discrete Applied Mathematics, vol.216, pp. 502–512, 2017.

[3] L. Aslanyan, H. Sahakyan, V. Romanov, G. Da Costa, and R. Kacimi, "Large network target coverage protocols", In 2019 Computer Science and Information Technologies (CSIT), pp. 57–64, IEEE, 2019.

[4] M. Azad, I. Chikalov, S. Hussain, M. Moshkov, B. Zielosko, "Decision Trees with Hypotheses for Recognition of Monotone Boolean Functions and for Sorting", In: Decision Trees with Hypotheses. Synthesis Lectures on Intelligent Technologies. Springer, Cham, 2022. https://doi.org/10.1007/978-3-031-08585-7_6.

[5] R. Bellman, "A Markovian decision process", Journal of Mathematics and Mechanics, vol.6, pp. 679–684, 1957. ISSN 0022-2518.

[6] A. Blum, "Learning a function of r relevant variables (open problem)", In Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop, pp. 731-733, 2003.

[7] E. Boros, P. L. Hammer, T. Ibaraki and K. Kawakami, "Identifying 2-monotonic positive Boolean functions in polynomial time, ISA'91, Algorithms, edited by W. L. Hsu and R. C. T. Lee, Springer Lecture Notes in Computer Science, vol. 557, pp. 104-115, 1991.

[8] N. Bshouty and Ch. Tamon, "On the Fourier spectrum of monotone functions". Journal of the ACM (JACM) vol. 43, no. 4, pp. 747-770, 1996.

[9] R. Dedekind, "Über Zerlegungen von Zahlen durch ihre größten gemeinsamen Teiler, Festschrift Hoch", Braunschweig u. ges. Werke, II, pp.103-148, 1897.

[10] P. Frankl and G.OH Katona, "On strengthenings of the intersecting shadow theorem", Journal of Combinatorial Theory, Series A, 184:105510, 2021.

[11] D. N. Gainanov, "On one criterion of the optimality of an algorithm for evaluating monotonic Boolean functions", Computational Mathematics and Mathematical Physics, vol. 24, pp. 176–181, 1984.

[12] M. Gara, T.S. Tasi, P. Balázs, "Machine Learning as a Preprocessing Phase in Discrete Tomography", Lecture Notes in Computer Science, vol. 7346, 2012.

[13] G. Katona, "A theorem of finite sets", In: Theory of Graphs. Academic Press - Akadémiai Kiadó, New York; Budapest, pp. 187-207, 1968.

[14] G. Katona, "A theorem of finite sets", In Classic Papers in Combinatorics, pp. 381–401. Springer, 2009.

[15] G. Hansel, "Sur le nombre des fonctions booléennes monotones de n variables", C. R. Acad. Sci., 262:1088–1090, 1966.

[16] Y. Hu, Y. Yao, W. Lee, "A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs", Knowl-Based Syst 204:106244, 2020.

[17] J.C. Jackson, , H.K.Lee, , R.A. Servedio, , A.Wan, "Learning random monotone DNF", In Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. APPROX RANDOM 2008. Lecture Notes in Computer Science, vol 5171. Springer, Berlin, Heidelberg, 2008. https://doi.org/10.1007/978-3-540-85363-3_38

[18] E. B. Khalil, H. Dai, Y. Zhang, B. Dilkina, L. Song. "Learning combinatorial optimization algorithms over graphs", Advances in Neural Information Processing Systems, 30:6351–6361, 2017.

[19] V. K. Korobkov, "On monotone functions of the algebra of logic", Problemy kibernetiki, vol.13, pp. 5–28, 1965.

[20] D. Korshunov, "On the number of monotone boolean functions", Problemy kibernetiki, vol. 38, pp. 5–109, 1981.

[21] J. Kruskal, "The number of simplices in a complex", Mathematical Optimization Techniques, University of California Press, pp. 251–278, 1963.

[22] Zh. Li, Q. Chen, and V. Koltun, "Combinatorial optimization with graph convolutional networks and guided tree search", Advances in neural information processing systems, vol. 31, 2018.

[23] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey", Computers and Operations Research, vol. 134, 2021.

[24] S. Muroga, "Threshold Logic and Its Applications", Wiley-Interscience, 1971.

[25] R. O'Donnell and R. Servedio, "Learning monotone functions from random examples in polynomial time", Citeseer, 2005.

[26] A. L. C. Ottoni, E. G. Nepomuceno, M.S. d. Oliveira, et al., "Reinforcement learning for the traveling salesman problem with Refueling", Complex & Intelligent Systems, 8, pp.2001–2015, 2022, https://doi.org/10.1007/s40747-021-00444-4.

[27] N. A. Sokolov, “On optimal deciphering of monotone functions of logic algebra”, Journal of Computational Mathematics and Math. Physics, 22(12), pp.1878–1887, 1982.
[28] N. A. Sokolov, “Optimal reconstruction of monotone Boolean functions” Computational Mathematics and Mathematical Physics, 27(6), pp.181–187, 1987.
[29] B.Schröder, “Ordered sets”. Springer 29 (2003): 30.
[30] R. S. Sutton and A.G. Barto, “Reinforcement learning”, The MIT Press, London, 2nd edition, 2018.
[31] G. P. Tonoyan, “Chain partitioning of n-cube vertices and deciphering of monotone Boolean functions”, Computational Mathematics and Mathematical Physics, 19(6), 1979.
[32] L. Valiant, “A theory of the learnable”, Communications of the ACM, 27(11), pp.1134-1142, 1984.
[33] O. Vinyals, M. Fortunato, N. Jaitly, “Pointer networks”, NeurIPS, pp. 2692–2700, 2015.
[34] H. Sahakyan, L. Aslanyan and V. Ryazanov, “On the Hypercube Subset Partitioning Varieties” 2019 Computer Science and Information Technologies (CSIT), Yerevan, Armenia, 2019, pp. 83-88, doi: 10.1109/CSITechnol.2019.8895211.
[35] K.Dohmen, “Improved Bonferroni Inequalities with Applications: Inequalities and Identities of Inclusion-Exclusion Type”. Berlin: Springer-Verlag, 2003.
[36] J. Galambos, J. and I. Simonelli, “Bonferroni-TypeInequalities with Applications”. New York: Springer-Verlag, 1996.

A Heuristic for Influence Maximization Under Deterministic Linear Threshold Model

Eszter Csókás, Tamás Vinkó
University of Szeged, Institute of Informatics, Hungary
E-mail: csokas@inf.u-szeged.hu, tvinko@inf.u-szeged.hu

Keywords: influence maximization, centrality metrics, deterministic linear threshold model

Received: April 16, 2023

Influence maximization (IM) is perhaps one of the most actively studied problems in network science. It is a combinatorial optimization problem in which, given a directed social network with influence weights, a spreading model, and a positive integer k, it is required to identify the set of seed nodes of size k which can make the largest influence in the network. We proposed an exact ILP model in our recent work and iterative solution approach to solve the IM problem under the so-called deterministic linear threshold spreading model. Since the solution describes how the diffusion happens for different time constraints, it is of interest to investigate how the various characteristics of the underlying graph relates to the result. In this paper, we present two new centrality metrics, computed from the input network structure, and use them to minimize the number of possible seed nodes. The solver chooses among the potential seed nodes and solves the problem, reducing the solution time. Benchmarking results are shown and discussed to demonstrate the efficiency of the proposed method.

Povzetek: Predlagana sta nova centralnostna kazalnika za optimizacijo izbire začetnih vozlišč v determinističnem linearnem modelu praga, kar izboljša učinkovitost in čas rešitve problema maksimizacije vpliva v omrežjih.

1 Introduction

Influence maximization Influence maximization (IM) is a combinatorial optimization problem. It studies a social network represented as a graph $G = (V, E, W)$, where V is the set of nodes in G, E is the set of directed edges in G and $W : E \to \mathbf{R}_+$ is a non-negative weight function. The goal of the problem is to find a $k \geq 1$ sized set of so-called seed nodes $v_1, \ldots, v_k \in V$ with the maximum influence in graph G in such a way that a weighted directed graph G, a diffusion (or spreading) model and the integer k are given [13].

The following notations will be used: $n = |V|$, for a node $j \in V$ the set of its out-neighbors is denoted by $N_{out}(j)$, and for $j \in V$ the set of in-neighbors is denoted by $N_{in}(j)$. Let $S \subset V$ of size $0 < k \leq n$ be the set of seeds and the function $\sigma(S)$ is the number of influenced nodes started from $S \subset V$ seeds by executing the diffusion model. The formal definition of the optimization problem is therefore

$$\max_{S \subset V, |S|=k} \sigma(S).$$

Diffusion models are usually used with stochastic parameters to solve influence maximization [13]. Thus, $\sigma(S)$ is the expected number of influenced nodes. The nodes with influenced and uninfluenced states will also be called as active and inactive nodes, respectively.

Diffusion models Several relevant diffusion models can be found in the literature, among which the independent cascade model [9], triggering model [13], time-aware model [16] and the linear threshold model [10] are the most popular ones. Our model is based on the linear threshold model (LTM) which solves the problem by iterating over a $t \in \mathbb{N}$ value, starting with $t = 1$. Let $0 < \theta_i < 1$ be the threshold value for each $i \in V$ that determines how easy it is to make the vertex active. The steps of the LTM is shown in Algorithm 1.

Algorithm 1 Linear threshold diffusion model

Step 1 Let $t = 1$, $0 < k \leq n$ be fixed and $\mathcal{V}_0$ be a seed set which containing k nodes, $\mathcal{V}_1 = \emptyset$.

Step 2 If

$$\sum_{j \in \tilde{N}(i)} b_{j,i} \geq \theta_i$$

holds for all $i \in V$ uninfluenced nodes, then put node i into the set $\mathcal{V}_t$. Mark those nodes as active at the end of this step.

Step 3 If there is no chance to influenced more node that is $\mathcal{V}_t = \emptyset$ holds, then STOP. $\mathcal{V} = \mathcal{V}_0 \cup \ldots \cup \mathcal{V}_{t-1}$ and $\sigma(\mathcal{V}_0) = |\mathcal{V}|$ by this time.

Otherwise, let $t := t + 1$ and go back to Step 2.

The value of σ in the LTM is determined by executing R runs using θ_i values uniform at random, then taking the average value of the influence to obtain the expected value of σ.

Deterministic linear threshold model In our work we used the deterministic linear threshold model (DLTM). It differs in that the θ_i values of the nodes are fixed, and thus no need for running the diffusion model multiple times. Note, that in [10] the original LT model is also deterministic. DLTM has been investigated also in the recent years, see e.g. in [1, 17, 12, 31, 19]. One of the key properties of DLTM is that the submodularity property does not hold [2], so there is no guarantee of the efficiency of the greedy algorithm.

Note that the IM under DLTM is a bilevel optimization problem since we need to find the maximum number of active nodes together with the minimum time t. Even using state-of-the-art optimization methods it is not possible to solve this kind of problem. Hence, an iterative solution method needs to be applied using the binary linear program which details are given in the next paragraph.

Integer LP model In the following, we present an integer linear programming (ILP) model, which is the result of our previous work [5]. This served as inspiration, since although it gives the global optimum of the problem, we only obtain results for small graphs within a reasonable run-time. Assuming that $\mathcal{T} > 1$ is a given integer constant and that $T = \{2, \ldots, \mathcal{T}\}$ is the set of time periods describing the diffusion process, the ILP model is as follows.

$$\max \sum_{i=1}^{n} x_{i,\mathcal{T}} \tag{1}$$

$$\sum_{i=1}^{n} x_{i,1} \leq k \tag{2}$$

$$\sum_{j \in N(i)} b_{j,i} x_{j,t-1} \geq \theta_i (x_{i,t} - x_{i,t-1}) \ \ \forall (i \in V, t \in T) \tag{3}$$

$$\sum_{j \in N(i)} b_{j,i} x_{j,t-1} \leq \theta_i + x_{i,t} - \varepsilon \ \ \forall (i \in V, t \in T) \tag{4}$$

$$x_{i,t-1} \leq x_{i,t} \ \ \forall (i \in V, t \in T) \tag{5}$$

$$\sum_{i=1}^{n} x_{i,\mathcal{T}-1} + 1 \leq \sum_{i=1}^{n} x_{i,\mathcal{T}} \tag{6}$$

$$\mathbf{x} \in \{0,1\}^{n \times \mathcal{T}} \tag{7}$$

The objective function (1) maximizes the number of active nodes in the last time epoch $\mathcal{T}$. Constraint (2) limits the number of seed nodes that can be initially selected. A node becomes active at a time t if the sum of the weights of the edges coming from its already active neighbours exceeds the threshold value of the vertex, defined by (3). Furthermore, the difference training on the right-hand side of this ensures that any node can be chosen as a seed at $t = 1$. This is related to the constraint (4) that the node is affected if the sum of the weights of the edges from the active neighbors exceeds the threshold of the vertex at time t. A small $\varepsilon > 0$ is required to ensure that the node is activated even if the sum of the weights of the edges arriving from the active neighbors equals the threshold value of the vertex[1]. The constraint (5) is that the state of the vertices cannot change backwards, i.e., once a node becomes active, it remains to be active. The equation (6) determines that at time $\mathcal{T}$ the number of influenced nodes should be greater than or equal to the number of active nodes at the preceding time in $\mathcal{T}-1$. Finally, constraint (7) describes that the solution matrix $\mathbf{x}$ is binary, i.e., one vertex is either influenced or uninfluenced at $t \in T$.

Centrality in general A centrality measure shows the nodes' importance in a graph which is based on the location of the nodes within the graph. Accurately, if given a graph $G(V, E, W)$, a centrality measure $C : V \to \mathbb{R}_+$ which assigns a non-negative centrality value to every node. The order of the nodes formed by the centrality values is usually more important than the centrality value itself [6, 27]. There are two main categories of centralities. One is based on shortest path, which includes closeness [26] and betweenness [8], among others. The other one is based on neighborhoods, where the centrality metrics, just to mention the most frequently used ones, are degree [28], eigenvector [3] and PageRank [4]. The centrality metric we have created is also based on neighborhoods what we describe below in Section 3.

2 Related work

The influence maximization is still an actively researched area of great interest and thus has a huge scientific literature. In this section, we would like to present the related works that are relevant, i.e., those based on the deterministic linear threshold model.

A fast algorithm for finding the most influential people was proposed in [22]. First, they were looking for communities of the graph and examining their limited number to reduce implementation time. The nodes to be excluded were selected using centrality measurements, community detection and new set computation. They then bounded the search space of the input network using the new set. The main advantage of the algorithm is that it reduces the number of nodes to be tested without compromising quality in order to reduce execution time.

The threshold-based greedy approach was presented in [14] for solving IM under DLTM. The greedy approach allows to obtain very good results compared to other combinatorial algorithms. It can be also used with the genetic algorithm and significantly improve its efficiency.

[1] An important note: we assume that the sum of the weights of the incoming edges is at most 1.

A new hybrid centrality metric based on closeness (harmonic) and decay measures was proposed in [29]. Two main application areas were presented. One hybridization was used to solve the coverage problem, while the other tries to find the most ideal node to reach the most people in the population, using the deterministic threshold model. In both cases, the centrality metric is based on a formulation of the weighted centrality measure of nodes.

The complexity of the IM under DLTM is studied in [18]. It is shown that for DLTM, active nodes are not approximable in $n^{1-\varepsilon}$ -factor polynomial time unless P=NP. In contrast, they are well approximable in the linear threshold model and the independent cascade model. It has also been demonstrated that for a given set of seeds, the number of influenced nodes can be determined in polynomial time.

The fact that the DLTM has no polynomial time $n^{1-\varepsilon}$ approximation unless P=NP, even when a person needs at most two active neighbors to become active was shown in [17]. However, there exists an $\varepsilon/(\varepsilon-1)$ polynomial-time approximation in the case where one of the neighbours has already become active and the person is can be activated. It is shown to be the best approximation under plausible Complexity-Theoretic assumptions.

In [11], the authors have created Targeted and Budgeted Influence Maximization for DLTM. They advanced a scalable algorithm that allows some optional methods to solve the problem, it is the TArgeted and BUdgeted Potential Greedy (TABU-PG) algorithm. It is an iterative and heuristics algorithm that relies on investing in potential future gains when choosing seed nodes.

In [30], the selection of top-k nodes is investigated based on the measure corresponding to the social network under consideration. It relies on the Shapley measure to efficiently compute an approximate solution to the problem. Although explicitly not using DLTM, the algorithm is general in a sense that it does not exploit the submodular property of the function.

3 New centralities

Two new centrality measures are proposed. Both of them are specifically developed for solving IM problem under DLTM. What makes them distinguished from other centralities is that they take into account not only the direction and the weight of the graph edges, but also the weight, i.e. threshold, of the nodes. To make it easier to understand our metrics and calculation, we have made a small graph, which is shown in Figure 1.

3.1 Influenceability

Our first centrality is the influenceability, denoted by $\mathcal{I}_{in}$, which measures how to easy activate a node. To calculate this, we examine the incoming edges from the neighbours, namely which edges and combinations of edges are able to reach or exceed the threshold value of the node. We define the weighted incidence, denoted by w_{to} as follows. Take the number of edge combinations that are able to activate the node by dividing by the number of edges in the edge combination and all occurrences of a given number of edge combinations. Finally, sum the w_{to} values and obtain $\mathcal{I}_{in}^{(p)} = \sum w_{to}(i)$, where $i \in V(G)$.

Table 1 shows the calculation of $\mathcal{I}_{in}^{(p)}$ for vertex 8 of the graph in Figure 1. The in-neighbors of vertex 8 are nodes $9, 4$ and 3. Note that vertex 4 can influence vertex 8 by itself, so combinations where vertex 4 is included will certainly be able to influence vertex 8.

Table 1: The $\mathcal{I}_{in}^{(p)}$ value of node 8 of the graph in Fig. 1.

combinations of edges	sum	θ	$w_{to}(8)$
$9 \to 8$	0.05	0.27	0
$4 \to 8$	0.27	0.27	$1/(1 \cdot 3)$
$3 \to 8$	0.18	0.27	0
$9 \to 8, 4 \to 8$	0.31	0.27	$1/(2 \cdot 3)$
$9 \to 8, 3 \to 8$	0.23	0.27	0
$4 \to 8, 3 \to 8$	0.44	0.27	$1/(2 \cdot 3)$
$9 \to 8, 4 \to 8, 3 \to 8$	0.49	0.27	$1/(3 \cdot 1)$
			$\mathcal{I}_{in}^{(p)}(8) = 1$

Table 2 shows the calculated $\mathcal{I}_{in}^{(p)}$ values for the vertices of the small graph in Fig. 1.

Table 2: The $\mathcal{I}_{in}^{(p)}$ values for the graph in Fig. 1.

node:	1	2	3	4	5	6	7	8	9
$\mathcal{I}_{in}^{(p)}$:	0	1	1	0	1	1.5	1.5	1	0

The final centrality metrics are obtained by combining with the measure of node and its neighbors. The influenceability value of a node is obtained by adding to the value of $\mathcal{I}_{in}^{(p)}$ the approximation of the influenceability of its in-neighbours:

$$\mathcal{I}_{in}(i) = \mathcal{I}_{in}^{(p)}(i) + \sum_{j \in N_{in}(i)} \frac{\mathcal{I}_{in}^{(p)}(j)}{|N_{out}(j)| - 1} \quad \forall (i \in V) \quad (8)$$

Note that if $|N_{out}(j)| \leq 1$, then let $|N_{out}(j)| = 2$ for the divisor to be 1.

3.2 Ability-to-influence

The second centrality is the ability-to-influence, denoted by $\mathcal{I}_{out}$. This indicates the influencing role of the node on its neighbors. Specifically, we look at all the combinations of incoming edges to the neighbourhood which include the edge from the investigated node. Of these, we count the ones whose sum of weights reaches the threshold value of the node and calculate the weighted incidence value for this case, denote w_{from}. As calculated for the influenceability, we divide the number of infecting edges by the number of edges in the edge combination and all occurrences of a

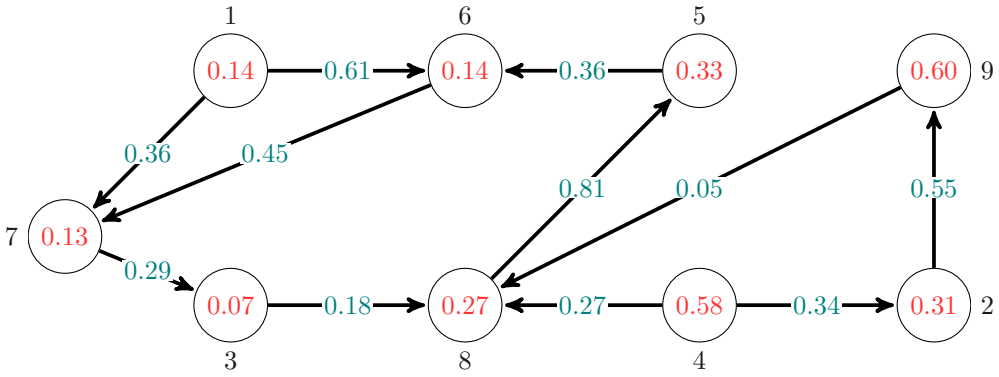


Figure 1: Example graph

given number of edge combinations. Finally, summarized w_{from} for each investigated combinations of edges.

The calculation of $\mathcal{I}_{out}^{(p)}$ value for node 4 is shown in Table 3. To do this, we need to look at the neighbours of node 4, i.e., the edges coming into the vertices 8 and 2. We have seen the combinations of edges coming into vertex 8 in Table 1, but only those that involve the edge coming from vertex 4 are needed.

To calculate this efficiently, we create a table with rows and columns representing the vertices of the graph. Row i and column j show the role of vertex j in the contamination of vertex i. The column sum of the table gives the $\mathcal{I}_{out}^{(p)}$ value of the vertices. We see this counting table in Table 4 which also shows the calculated $\mathcal{I}_{out}^{(p)}$ values for the vertices of the small graph in Fig. 1.

Table 3: The $\mathcal{I}_{out}^{(p)}$ value for node 4 of the graph in Fig. 1.

combinations of edges	sum	θ	$w_{from}(4)$
$4 \to 8$	0.27	0.27	$1/(1 \cdot 1)$
$9 \to 8, 4 \to 8$	0.31	0.27	$1/(2 \cdot 2)$
$4 \to 8, 3 \to 8$	0.44	0.27	$1/(2 \cdot 2)$
$9 \to 8, 4 \to 8, 3 \to 8$	0.49	0.27	$1(3 \cdot 1)$
$4 \to 2$	0.34	0.31	$1/(1 \cdot 1)$
			$\mathcal{I}_{out}^{(p)}(4) = 2.83$

The ability-to-influence value of a vertex is obtained by adding to the value of $\mathcal{I}_{out}^{(p)}$ the approximation of the ability-to-influence of its out-neighbors:

$$\mathcal{I}_{out}(i) = \mathcal{I}_{out}^{(p)}(i) + \sum_{j \in N_{out}(i)} \frac{\mathcal{I}_{out}^{(p)}(j)}{|N_{in}(j)| - 1} \quad \forall (i \in V) \quad (9)$$

Note that if $|N_{in}(j)| \leq 1$, then let $|N_{out}(j)| = 2$ for the divisor to be 1.

3.3 Potential seed selection

Using the two centrality values, we want to determine which vertices can be seeds. Therefore, first, the centrality values are normalized between 0 and 1 in a way that all the elements are divided with the maximum. Such normalization is denoted in each case by $||.||$. Then, we sort the nodes according to their centrality value. We put them in descending order according to their ability-to-influence value, since seed vertices should have good ability-to-influence's value. Conversely, we rank the vertices in ascending order according to their influenceability value, since seed vertices are unlikely to be easily infected. We take the weighted sum of the two order values for each node to get $\mathcal{I}$. This is shown in equation (10):

Table 4: The $\mathcal{I}_{out}^{(p)}$ values for the graph in Fig. 1.

	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	1.00	0	0	0	0	0
3	0	0	0	0	0	0	1.00	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1.00	0
6	1.50	0	0	0	1.50	0	0	0	0
7	1.50	0	0	0	0	1.50	0	0	0
8	0	0	0.58	1.83	0	0	0	0	0.58
9	0	0	0	0	0	0	0	0	0
$\mathcal{I}_{out}^{(p)}$	3.00	0.00	0.58	2.83	1.50	1.50	1.00	1.00	0.58

$$\mathcal{I}(i) = \alpha \cdot \text{ord}(||\mathcal{I}_{out}(i)||) + (1-\alpha) \cdot \text{ord}(||\mathcal{I}_{in}(i)||), \quad (10)$$

for all $i \in V$.

The normalized values of influenceability and ability-to-influence for the graph in Figure 1 are shown in Table 5. Also, the $\mathcal{I}$ values are calculated from them.

Finally, to form the set of potential seed nodes, choose the subset of $V(G)$ according to $\mathcal{I}$. This is controlled by a parameter $0 < r < 1$, thus the cardinality of the candidate seeds set is $r \cdot |V(G)|$.

Table 5: The value of $||\mathcal{I}_{in}||$, $||\mathcal{I}_{out}||$ and $\mathcal{I}(i)$ with $\alpha = 0.0$ and $\alpha = 0.8$ for the graph in Figure 1.

node	$\|\|\mathcal{I}_{in}\|\|$	$\|\|\mathcal{I}_{out}\|\|$	$\mathcal{I}(i), \alpha = 0.0$	$\mathcal{I}(i), \alpha = 0.8$
1	0.0	1.0	1	1.0
2	0.33	0.11	3	7.8
3	0.83	0.20	7	7.0
4	0.0	0.61	2	2.0
5	0.67	0.55	5	3.4
6	0.83	0.45	8	4.8
7	1.0	0.29	9	6.6
8	0.67	0.45	6	5.2
9	0.33	0.20	4	7.2

4 Algorithms

Before details are given about our new heuristic method it needs to be emphasized that due to the nature of our problem, namely finding the maximum number of influenced nodes with minimum diffusion time steps, all algorithms work iteratively and use the ILP model $(1) - (7)$ and their stopping criteria is to run until infeasibility. This usually leads to a long running time.

4.1 Proposed heuristic: IAtI

Here we describe our proposal for a heuristic which selects a candidate seeder set of graph nodes based on the new centrality metrics introduced in Section 3. Since it is using the Influenceability and the Ability-to-Influence measures we refer to it as IAtI-heuristic.

The method is described in Algorithm 2. In its precondition phase, in Step 1 and 2 it calculates the two centrality metrics and the combination of them using the formulae given in Section 3. In Step 3 the algorithm collects the set of possible seed nodes. It is a parameter $0 < r < 1$ which controls the ratio of the nodes to be selected. Higher r value leads to higher probability for the seed nodes corresponding to the global optimum to be selected into the candidate set. However, high r value also leads to higher execution time, thus it needs to be set up with care. Steps $4 - 6$ describe the iterative part of the algorithm. This is essentially the same as the algorithm we proposed in our recent work [5]. Note that the algorithm usually iterates until the ILP model becomes infeasible, as there is no other stopping criteria, unless all the nodes become active; the reader is referred to [5] for the theoretical proofs.

Discussion on parameters' choice for IAtI Now that we have Algorithm 2, let us see how to set up its parameters in order to have high chance to include the seed nodes of the global optimum and exclude those which might not be good candidates. As we have calculated the centrality metrics for the small test graph from Figure 1 as well as the globally optimal solution using $k = 2$ seed nodes, the scatterplot shown in Figure 2 clearly suggests that we shall aim

Algorithm 2 IAtI-heuristic(r, α)

Input A directed graph G with edge weights and node threshold values.

Step 1 Calculate $\mathcal{I}_{in}^{(p)}$ and $\mathcal{I}_{out}^{(p)}$ for all $i \in V$ and then $\mathcal{I}_{in}$ and $\mathcal{I}_{out}$ using Eq. (8) and (9), respectively.

Step 2 Form $\mathcal{I}$ for each vertex according to the Equation (10) using the input parameter α.

Step 3 Define $S \subseteq V(G)$ to be the set of possible seeds: choose the top $r \cdot |V(G)|$ number of nodes from $\mathcal{I}$.

Step 4 Let $\mathcal{T} := 2$ and start the iteration.

Step 5 Solve the ILP defined by $\{(1)$ - $(7)\}$ for the diffusion time value $\mathcal{T}$, so that the seed vertices can be chosen exclusively from the set S.

Step 6 If all the nodes are influenced or the solution becomes infeasible then stop the iteration. Otherwise, let $\mathcal{T} = \mathcal{T} + 1$ and go back to Step 5.

for larger ability-to-influence value together with low influenceability. The explanation is that the seed nodes of the global optimum are colored red (node 4 and 1). Larger circle represents later activation in the diffusion in the globally optimal solution[2]. Obviously, this landscape of the centrality values is quite simple as our small test graph is of special structure. Discussion some larger graphs will be given in Section 5.3.

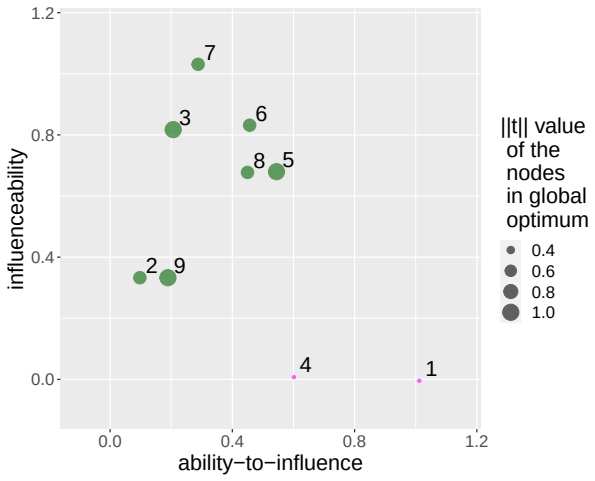


Figure 2: Visualization of the new centralitites for the small test graph

[2]It must be emphasized here that the size of the circles on Figure 2 represents the activation time in the globally optimal solution. For non-optimal $\mathcal{T}$ value we might obtain different times.

Two different α values were used from which potential seeds were selected in equal proportions. In the first case, the weight of the ability-to-influence property is set to a higher value, so $\alpha = 0.8$ and select elements with a ratio $r = 0.2$ which are the best elements according to $\mathcal{I}$. Moreover, we add the vertices with ability-to-influence value equal to 1. This is necessary because it has the highest chance of infection, yet in many cases it will not be the seed. There could be several reasons for this, for example another node could reach the same global optimum or such a node could be a neighbour of the seed that could infect the seed. In any case, choosing these nodes gives a better chance for finding the global optimum. Denote this set by S_1. In the second case, we select based on the influenceability value alone, so $\alpha = 0$ and added nodes with 0 influenceability value. This is because nodes with an influence value of 0 can never be influenced unless they become seed nodes themselves. Therefore, any vertex with an influence value of 0 is chosen to be a seed vertex. The resulting set is denoted by S_2. Finally, take the intersection of S_1 and S_2 and obtain the set S. The seed nodes are chosen from this set S.

4.2 Greedy

Although the so-called submodularity property does not hold for the DLTM, the greedy approach is still a favorable method for solving the IM problem. We have adapted the greedy strategy into our ILP framework. For the formal description of the method the reader is referred to the Algorithm 3 in the Appendix.

5 Numerical experiments

5.1 Computational environment

The implementation of the above proposed new centralities was done in R version 4.1.2 using its `igraph` 1.3.5 package. The numerical experiments were executed with Gurobi 10.0 called from AMPL [7] using non-default options: `threads=8 lpmethod=0 cuts=0 mipgapabs=1e-2`. The computer used had Intel Core i7-10700 CPU at 2.90GHz with 16GB memory running Ubuntu 22.04.2. Note that we used multi-core setup of Gurobi (i.e., for solving the ILP models), whereas R was used with single threading. In the R implementation of the new centrality metrics we used the different `apply` functionalities to process matrices and lists efficiently.

5.2 Test graphs

Synthetic random graphs To generate random test graphs we used the LFR scheme, which creates networks with prescribed community structures [15]. Note that this procedure only provides the graphs of social-network type, the edge weights and node thresholds needed to be assigned in the second phase. Similarly to our earlier work [5], the following procedure was used.

Table 6: A selection of real-world test graphs

name	N	M	$d_{\max}$
soc-dolphins [20]	62	159	5
ca-sandi-auths	86	124	12
retweet [25, 24]	96	117	17
ca-netscience [21]	379	914	34

– Regarding the edge weights: nodes with in-weights larger than 1 (given by the LFR method) were normalized to 1; and we applied a multiplication with a factor r_w which was a uniform at random number in the interval $[0.6, 1]$.

– The nodes' threshold values were generated uniform at random in the interval $[0.05, 0.6]$.

– From the LFR method three parameters were fixed: mixing parameter $\mu_w = 0.1$, minimal community size $minc = 5$, maximal community size $maxc = 42$.

– Three parameters were varied in the experiments: number of nodes n, the average degree $avgk$, and maximum degree $maxk$.

Real-world graphs We have tested our proposed method on a small set of real-world graphs [23], see Table 6 for the details. Note that the last column indicates the maximum degree of the given graph. For our problem setup $d_{\max}$ needs to be relatively small, since the exact ILP solver as well as our heuristic does not scale efficiently. Nevertheless, for these four social networks we did the following experiments.

The real graphs were undirected and unweighted graphs. The `mutual` parameter was used which makes the graphs directed by doubling the undirected edges. We then created 3 groups according to the percentage of edges that were deleted randomly and how we generated weight of edges and threshold values. These groups and their corresponding generating parameters are shown in Table 7. The edges and weights were generated uniform at random in the given interval. As for the weights of the edges, similar to the LFR graphs, nodes with weights greater than 1 were normalized to 1 and multiplied by a factor r_w which were randomly chosen from the interval $[0.6, 1]$.

Table 7: The parameters used to generate real-world test graphs

group	% of edges deleted	threshold	weights
#1	45	$[0.05, 0.5]$	$[0.01, 0.6]$
#2	50	$[0.1, 0.5]$	$[0.05, 0.5]$
#3	50	$[0.07, 0.55]$	$[0.075, 0.55]$

5.3 Results

For all the test graphs we ran the new IAtI-heuristic and compare its results with those obtained by Greedy. For reference we also solved the smaller sized problems with Gurobi to get the global optimal solution. Note that we fixed the seed set size to 2.

LFR graphs The results are reported in Table 8 (including the globally optimal solution) and 9 (without global optima[3]). If the obtained results (with respect to σ and $\mathcal{T}$) were different, then the better solution is emphasized as boldface. For the smaller sized problems we can see that there were 3 cases when neither our heuristic nor the greedy approach was able to find the global optimum, see Table 8. Greedy found better solutions than our heuristic in seven cases. Also in 7 cases (that is 23%) our heuristic missed the global optimum. On the other hand, Greedy lost the global optimum in 11 cases (37% of the cases). Regarding the running time, the greedy approach is far the quickest method. On the other hand, IAtI-heuristic is faster that the exact method, usually around 4−12 times.

Table 8: Benchmarking results for the LFR graphs; time in seconds

graph parameters			IAtI-heuristic			greedy			global opt.		
n	$avgk$	$maxk$	σ	$\mathcal{T}$	time	σ	$\mathcal{T}$	time	σ^*	$\mathcal{T}^*$	time
100	3	8	87	13	22.2	87	13	11.9	87	13	164.2
100	4	8	74	17	77.1	74	17	7.6	74	17	1442.3
100	4	10	100	15	11.7	100	15	5.5	100	15	20.6
100	5	8	88	24	339.1	88	24	12.4	88	24	4287.9
100	5	10	89	10	168.7	89	10	5.9	89	10	1742.7
100	6	8	88	17	348.1	88	17	9.6	88	17	4353.3
105	3	8	58	11	8.9	58	11	5.4	58	11	92.0
105	4	8	57	10	15.5	57	10	2.4	57	10	169.0
105	4	10	**80**	**15**	16.0	79	14	6.2	80	15	42.2
105	5	8	85	16	20.1	85	16	7.4	85	16	86.9
105	5	10	66	20	41.7	66	20	8.7	66	20	108.0
105	6	8	61	14	27.7	61	14	4.8	**61**	**12**	119.5
110	3	8	75	20	110.0	75	20	10.7	**75**	**15**	655.8
110	4	8	**96**	**23**	411.9	87	15	9.4	96	23	5510.2
110	4	10	91	16	20.9	91	16	10.1	91	16	188.0
110	5	8	90	18	38.9	90	18	9.4	90	18	313.7
110	5	10	70	15	236.5	70	15	7.1	70	15	430.7
110	6	8	**95**	**15**	221.0	95	19	16.6	95	15	1085.0
115	3	8	56	13	6.6	56	13	4.6	56	13	59.3
115	4	8	49	18	13.5	45	14	6.4	49	18	170.5
115	4	10	43	11	12.3	**49**	**13**	4.2	49	13	64.3
115	5	8	**25**	**18**	24.9	21	7	1.1	25	18	219.5
115	5	10	**78**	**20**	81.7	68	14	6.3	78	20	193.1
115	6	8	**49**	**13**	18.2	46	11	4.3	49	13	292.7
120	3	8	86	26	59.7	**86**	**25**	23.4	86	25	1402.6
120	4	8	**77**	**17**	30.9	75	15	6.9	77	17	307.2
120	4	10	113	20	36.4	**113**	**14**	18.2	113	14	274.9
120	5	8	70	12	97.0	70	14	9.6	**70**	**11**	469.5
120	5	10	82	11	67.6	**83**	**17**	9.9	83	17	326.7
120	6	8	**98**	**15**	113.7	98	17	12.5	98	15	1160.0

For the larger graphs, reported in Table 9 we can see that the in terms of running time Greedy is much faster than our heuristic. Greedy was only 7 cases found better solution than our heuristic. On the other hand IAtI-heuristic was in 13 cases more successful than the greedy approach and among these cases there are several ones where it obtained much better solution.

[3]Due to the long running time for the larger graphs the globally optimal solutions are not available.

Table 9: Benchmarking results for the LFR graphs, globally optimal solution not available; time in seconds

graph parameters			IAtI-heuristic			greedy		
n	$avgk$	$maxk$	σ	$\mathcal{T}$	time	σ	$\mathcal{T}$	time
125	3	8	116	17	35.5	116	17	13.4
125	4	8	102	24	407.1	102	24	25.6
125	4	10	93	22	115.0	**98**	**27**	25.1
125	5	8	81	15	303.7	81	15	16.9
125	5	10	99	19	220.9	**99**	**18**	29.4
125	6	8	**104**	**13**	406.1	104	14	15.2
130	3	8	121	28	604.7	121	28	90.4
130	4	8	**96**	**14**	266.5	84	10	11.7
130	4	10	**51**	**9**	80.6	51	10	5.9
130	5	8	130	20	116.5	**130**	**19**	16.3
130	5	10	114	16	355.2	**116**	**16**	27.1
130	6	8	130	15	54.2	130	15	11.9
135	3	8	129	19	22.7	**135**	**18**	10.6
135	4	8	117	13	215.9	117	13	22.7
135	4	10	93	14	65.4	93	14	15.8
135	5	8	**74**	**21**	864.4	55	16	12.1
135	5	10	81	13	234.5	81	13	15.5
135	6	8	**113**	**23**	1059.6	97	18	36.3
140	3	8	57	11	17.6	57	11	18.4
140	4	8	**116**	**20**	247.9	114	16	13.3
140	4	10	82	17	32.5	82	17	14.0
140	5	8	**93**	**18**	369.2	72	16	22.7
140	5	10	111	17	172.8	111	17	21.9
140	6	8	**130**	**32**	1283.5	103	17	17.5
145	3	8	70	16	34.6	70	16	14.0
145	4	8	100	21	273.4	100	21	45.2
145	4	10	60	18	123.8	**76**	**19**	17.2
145	5	8	**58**	**20**	134.7	44	14	7.8
145	5	10	**110**	**21**	621.5	108	20	30.0
145	6	8	101	25	885.1	101	25	31.1
150	3	8	72	12	107.1	72	12	17.2
150	4	8	66	13	167.3	**66**	**12**	9.8
150	4	10	**84**	**16**	78.6	83	17	13.9
150	5	8	**123**	**21**	290.5	118	19	28.2
150	5	10	89	27	1067.1	89	27	61.5
150	6	8	**130**	**22**	1207.6	124	18	20.6

To discuss a particular example at which the IAtI-heuritic lost the globally optimal solution, whereas Greedy was able to find it, see Figure 3. The scatterplot show the two centralities of the nodes together with their activation time at the optimal $\mathcal{T}^*$. The empty circles correspond to those nodes which were not selected by our heuristic to be candidates for seeds. As shown by the red circles, we can see that IAtI dropped the optimal seed set, as only those were selected to be candidates which are shown with dark-blue color. Note that Greedy, instead, found those red-colored nodes. The seed nodes chosen by the IAtI-heuristic are shown as yellow colored circles. It would be possible to parameterize the IAtI-heuristic in such a way that it would include the optimal seeds in the candidate set. Figure 3 suggests that in that case the cardinality of the candidate seed set would be necessary larger which would result in much longer running time.

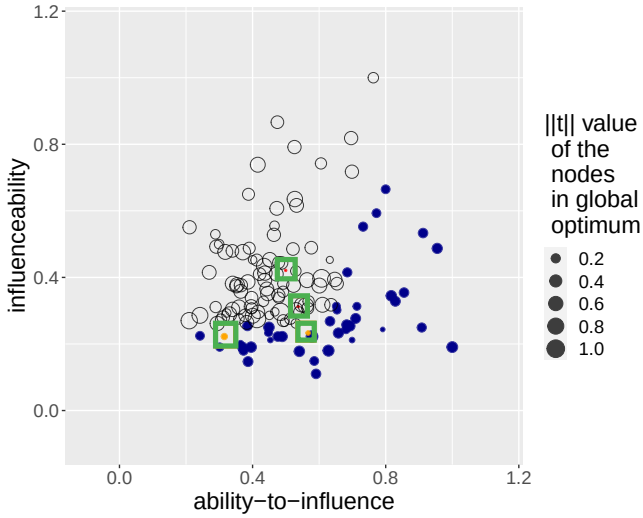


Figure 3: Visualization of the new centralitites for the LFR.135.3.8 graph

Real-world graphs Our experiments on some selected real world graphs are reported in Table 10. As in the earlier tables it is shown in boldface if a method obtained a better solution than the other one. We can observe that the trend of the IAtI-heuristic being much slower than Greedy remains to be the case also here. There was only once case when the greedy approach found a better solution than our heuristic. On the other hand, IAtI-heuristic was able to find better solution in 5 cases (out of 12), which corresponds to 42%.

Table 10: Results for the small real-world graphs; time in second

	IAtI-heuristic			greedy		
graph	σ	$\mathcal{T}$	time	σ	$\mathcal{T}$	time
soc-dolphins #1	**45**	**18**	11.9	41	10	5.1
ca-sandi-auths #1	**34**	**9**	6.9	30	9	1.7
retweet #1	22	7	5.5	22	7	0.9
ca-netscience #1	63	9	319.5	63	9	183.6
soc-dolphins #2	**34**	**9**	6.9	24	5	0.5
ca-sandi-auths #2	**13**	**5**	5.5	12	4	0.3
retweet #2	19	6	4.8	19	6	0.7
ca-netscience #2	25	7	3047.7	**25**	**6**	13.8
soc-dolphins #3	27	8	5.5	27	8	1.5
ca-sandi-auths #3	14	5	7.6	14	5	0.5
retweet #3	17	6	4.9	17	6	0.6
ca-netscience #3	**41**	**11**	300.1	30	5	10.2

Finally, let us demonstrate again the seed node selection strategy of the IAtI-heuristic on the ca-sandi-auths graph as shown in Figure 4. The red colored circles represent the nodes found by the Greedy approach, leading to a suboptimal result. The red circle on the left was exluded by the IAtI-heuristic from the possible set of seed nodes. As before, the candidate set of seed nodes are colored with dark-blue. Among them, there are the two optimal seed nodes shown as yellow circles. Note that there are quite many nodes with 0 influenceability value. Those are definitely selected by IAtI as possible seed nodes as they are impossible to be activated by other nodes. As we can observe, one of them is indeed part of the optimal seed set.

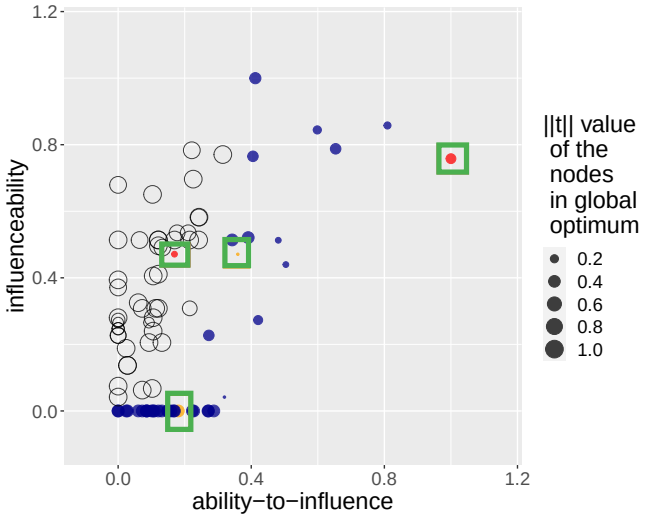


Figure 4: Visualization of the new centralitites for the ca-sandi-auths #1 real graph

6 Conclusion

We proposed two new centrality metrics for the influence maximization under deterministic linear threshold model. These metrics take into account the structure of the input network, more precisely the weight of edges, the combinations of edges and the threshold value of vertices. This is a great advantage of our method, because it is not usual to include the node's threshold in the centrality measure. Using the two centrality metrics, we selected vertices that have a high probability of being seed nodes. The solver now selects seed vertices only among these. This reduces the computational complexity of the task and therefore, compared to running the ILP solver on the unrestricted model, it speeds up the procedure. Using those metrics, we created the so-called IAtI algorithm. This was compared with Greedy and in some cases with the global optimum, also. The IAtI algorithm is slower than Greedy, but in many cases it gives a better solution and in most cases it finds the global

optimum. Although most real-world networks are sparse graphs, for which our method works well, the disadvantage of this method is that it takes a lot of computing time to generate and use edge combinations for large degree of nodes. Thus, it may take a long time for pre-processing and is therefore, in its current form, not scalable. Our future work is to develop a version of the method that can handle graphs including nodes with relatively large degree.

Acknowledgement

The research leading to these results has received funding from the national project TKP2021-NVA-09. Project no. TKP2021-NVA-09 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development, and Innovation Fund, financed under the TKP2021-NVA funding scheme. The work was also supported by the grant SNN-135643 of the National Research, Development, and Innovation Office, Hungary.

References

[1] D. Acemoglu, A. Ozdaglar, and E. Yildiz. Diffusion of innovations in social networks. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 2329–2334, 2011. https://doi.org/10.1109/cdc.2011.6160999

[2] F. Altarelli, A. Braunstein, L. Dall'Asta, and R. Zecchina. Optimizing spread dynamics on graphs by message passing. *Journal of Statistical Mechanics: Theory and Experiment*, page P09011, 2013. https://doi.org/10.1088/1742-5468/2013/09/p09011

[3] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology*, 2:113–120, 1972. https://doi.org/10.1080/0022250x.1972.9989806

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998. https://doi.org/10.1016/s0169-7552(98)00110-x

[5] E. Csókás and T. Vinkó. An exact method for influence maximization based on deterministic linear threshold model. *Central European Journal of Operations Research*, 31:269–286, 2023. https://doi.org/10.1007/s10100-022-00807-3

[6] K. Das, S. Samanta, and M. Pal. Study on centrality measures in social networks: a survey. *Social Network Analysis and Mining*, 8, 2018. https://doi.org/10.1007/s13278-018-0493-2

[7] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL. A modeling language for mathematical programming*. Thomson, 1993. https://doi.org/10.1007/978-3-642-83724-1_12

[8] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977. https://doi.org/10.2307/3033543

[9] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001. https://doi.org/10.1023/a:1011122126881

[10] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83:1420–1443, 1978. https://doi.org/10.1086/226707

[11] F. Gursoy and D. Gunnec. Influence maximization in social networks under deterministic linear threshold model. *Knowledge-Based Systems*, 161:111–123, 2018. https://doi.org/10.1016/j.knosys.2018.07.040

[12] P.D. Karampourniotis, B.K. Szymanski, and G. Korniss. Influence maximization for fixed heterogeneous thresholds. *Scientific Reports*, 9:5573, 2019. https://doi.org/10.1038/s41598-019-41822-w

[13] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003. https://doi.org/10.1145/956750.956769

[14] S. Kochemazov. Comparative study of combinatorial algorithms for solving the influence maximization problem in networks under a deterministic linear threshold model. *Procedia Computer Science*, 136:190–199, 2018. 7th International Young Scientists Conference on Computational Science, YSC2018. https://doi.org/10.1016/j.procs.2018.08.252

[15] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78:046110, 2008. https://doi.org/10.1103/physreve.78.046110

[16] B. Liu, G. Cong, D. Xu, and Y. Zeng. Time constrained influence maximization in social networks. In *2012 IEEE 12th International Conference on Data Mining*, pages 439–448, 2012. https://doi.org/10.1109/icdm.2012.158

[17] Z. Lu, W. Zhang, W. Wu, B. Fu, and D. Du. Approximation and inapproximation for the influence maximization problem in social networks under deterministic linear threshold model. In *2011 31st International Conference on Distributed Computing Systems*

Workshops, pages 160–165, 2011. https://doi.org/10.1109/icdcsw.2011.33

[18] Z. Lu, W. Zhang, W. Wu, and J. Kim. The complexity of influence maximization problem in the deterministic linear threshold model. *Journal of Combinatorial Optimization*, 24:374–378, 2012. https://doi.org/10.1007/s10878-011-9393-3

[19] Z. Lu, W. Zhang, W. Wu, J. Kim, and B. Fu. The complexity of influence maximization problem in the deterministic linear threshold model. *Journal of Combinatorial Optimization*, 24(3):374–378, 2012. https://doi.org/10.1007/s10878-011-9393-3

[20] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003. https://doi.org/10.1007/s00265-003-0651-y

[21] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006. https://doi.org/10.1103/physreve.74.036104

[22] K. Rahimkhani, A. Aleahmad, Ma. Rahgozar, and A. Moeini. A fast algorithm for finding most influential people based on the linear threshold model. *Expert Systems with Applications*, 42:1353–1361, 2015. https://doi.org/10.1016/j.eswa.2014.09.037

[23] R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. https://doi.org/10.1609/aaai.v29i1.9277

[24] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, and M. A. Patwary. What if clique were fast? maximum cliques in information networks and strong components in temporal networks. *arXiv preprint arXiv:1210.5802*, pages 1–11, 2012. https://doi.org/10.48550/arXiv.1210.5802

[25] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, and M. A. Patwary. Fast maximum clique algorithms for large graphs. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*, 2014. https://doi.org/10.1145/2567948.2577283

[26] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966. https://doi.org/10.1007/bf02289527

[27] S. Segarra and A. Ribeiro. Stability and continuity of centrality measures in weighted graphs. *IEEE Transactions on Signal Processing*, 64:543–555, 2016. https://doi.org/10.1109/tsp.2015.2486740

[28] M. E. Shaw. Group structure and the behavior of individuals in small groups. *The Journal of Psychology*, 38:139–149, 1954. https://doi.org/10.1080/00223980.1954.9712925

[29] A. Singh, R. Singh, and S. Iyengar. Node-weighted centrality: a new way of centrality hybridization. *Computational Social Networks*, 7:Article nr. 6, 2020. https://doi.org/10.1186/s40649-020-00081-w

[30] N. Rama Suri and Y. Narahari. Determining the top-k nodes in social networks using the Shapley value. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, pages 1509–1512, 2008. https://dl.acm.org/doi/10.5555/1402821.1402911

[31] R. Xu. An L_p norm relaxation approach to positive influence maximization in social network under the deterministic linear threshold model. In *Algorithms and Models for the Web Graph*, pages 144–155. Springer, 2013. https://doi.org/10.1007/978-3-319-03536-9_12

Appendix

The description of the greedy approach is given in Algorithm 3.

Algorithm 3 Greedy algorithm using the ILP model

Step 1 Let $k := 1$ be the number of seed nodes.

Step 2 Let $\mathcal{T} := 2$ and start the iteration with diffusion time.

Step 3 Solve the ILP defined by $\{(1)\text{-}(7)\}$ for the diffusion time value $\mathcal{T}$.

Step 4 If all the nodes are influenced or the model becomes infeasible then the optimum is found and stop the iteration with $\mathcal{T}$ and go to Step 5. Otherwise, let $\mathcal{T} = \mathcal{T}+1$ and go back to Step 3.

Step 5 If $k = |S|$ where S is set of seeds, then the stop the algorithm. Otherwise, fix the selected seed nodes for the remaining iterations. Let $k := k+1$ and go to Step 2.

https://doi.org/10.31449/inf.v48i4.4752 Informatica **48** (2024) 543–548 **543**

Empirical Evaluation of Algorithm Performance: Addressing Execution Time Measurement Challenges

Tomaž Dobravec
Faculty of Computer and Information Science
University of Ljubljana, Slovenia
E-mail: tomaz.dobravec@fri.uni-lj.si

Keywords: Empirical algorithm analysis, time measuring, accuracy, reliability, Java, C, x86 assembly

Received: March 21, 2023

In this paper, we investigate the influence of various factors, such as programming language, testing environment, and input data, on the accuracy of algorithm execution measurements. To conduct this study, we used the BubbleSort algorithm as a test case and implemented it in Java, C, and x86 assembly languages. We executed these implementations with various inputs and performed an empirical evaluation of the results using the ALGator system. We showed that the influence of the chosen programming language is negligible, since the Java and C implementations gave very similar results, while the assembler implementation differed only by a constant factor. Furthermore, our analysis emphasized the importance of repeating tests to obtain precise timing measurements - the more tests we do, the more accurate the measured result will be. We also discuss the impact of the input data type which can significantly affect the execution time due to the increased number of mispredictions of the branch predictor.

Povzetek: Narejena je empirična študija vpliva programskega jezika, ponavljanja testov in tipa vhodnih podatkov na točnost meritev časa izvajanja algoritmov.

1 Introduction

The analysis of algorithmic complexity is a crucial component of the algorithm design process [3]. This analysis is primarily concerned with estimating the amount of resources (such as time or memory usage) that an algorithm will require during its execution [10]. The outcome of this analysis is dependent upon the chosen computation model [4], which encompasses the execution environment and its limitations. Generally, the results of theoretical complexity analysis are used to differentiate between fast (i.e., polynomial) and slow (exponential) algorithms. However, the practical value of these results is limited, especially when two algorithms have the same (theoretical) time complexity. This is because the theoretical model used in the analysis does not take into account all the intricacies of the actual execution environment, such as memory caching, paging, or branch prediction [6, 7], which are only revealed during the execution of the algorithm on a real computer. Therefore, in order to make a practical comparison of algorithms, the theoretical analysis must be supplemented with empirical measurements of resource utilization during algorithm execution on various input data types [5, 8, 11].

To obtain accurate and reliable results, these measurements must be performed with great care, as numerous factors can impact the data being measured. This paper focuses on some of these factors and presents the results of our measurements that highlight their significance. Specifically, we employ three programming languages and demonstrate the impact of language selection on the speed of execution. Furthermore, we emphasize the importance of repeating tests, especially when the size of the input (and consequently, the execution time) is small. In addition, we discuss how the input data type can affect the rankings of algorithm quality. By thoroughly examining these factors and presenting our results, we hope to contribute to a more comprehensive understanding of the practical implications of algorithmic complexity analysis.

2 Testing environment setup

For all our tests in this research we will use the BubbleSort [1] algorithm for sorting arrays of integers. Since this is a very well-known and simple algorithm we are able to perform a precise theoretical analysis and provide a very accurate (theoretical) forecast for the time complexity of its implementations. The algorithm is so simple that we can count the number of operations performed during the execution for different inputs. Thus we will be able to compare theoretical predictions with the empirical results.

One of the goals of this research was to analyze the impact of the selected programming language on the efficiency of algorithm execution. Therefore we used three programming languages (namely Java, C, and the x86 assembler) to implement BubbleSort. Due to the simplicity of the algorithm, we managed to write the three implementations in such a way that they provide semantically identical code (see listings in Fig. 1). For further reference, we

named implementations `BubbleJ`, `BubbleC` and `BubbleA`, where the last letter denotes the programming language used (J for Java, C for C, and A for x86 assembler). Executing these implementations on the same inputs will result in an equal number of each programming-language-dependant atomic operations. Any differences in the execution speed will thus reflect the differences in the execution speed of these operations in the selected programming language. The C implementation was compiled with the `gcc` compiler in two ways: without optimization (the `-O0` flag) and with full optimization (the `-O3` flag). In this way, we got two distinct implementations (namely `BubbleC0` and `BubbleC3`). In the following, we will analyze the impact of this optimization on the speed of execution.

To facilitate the empirical evaluation in our research we used the ALGator system [2]. We used its tools to configure the Sorting project, to provide the test sets of input data and implementations, and to execute the algorithms' implementations in a controlled environment. For the execution machine, we used the Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz computer with 32GB RAM and with the Linux Ubuntu operating system installed.

The inputs for our algorithms consist of arrays of integers prearranged in three different orders: random order (RND), sorted order (SOR), and inversely sorted order (INV). These three distributions of input data are well manageable from a theoretical point of view since we know for all three the number of operations that will be performed during the sorting process. In all three case BubbleSort will perform exactly $n(n-1)/2$ comparisons, and $n^2/4$, 0, $n(n-1)/2$ swaps for RND, SOR, INV respectively. Note that all the numbers of operations are exact, except for the number of swaps in the RND case - here we only have the expected (instead of exact) number of swaps, since the sequence is randomly mixed. Since BubbleSort performs only comparisons and swaps (and some auxiliary increments of indices to maintain the loops) we could expect that, for example, sorting the RND array will be faster than sorting the INV array of the same size. But as we will see in the following this is not the case.

In the ALGator project inputs (i.e. test cases) are grouped into test sets. Each test case has its own identifier (Test ID), so the results can also be compared on a test basis. To provide accurate results each test case is executed several times (each execution of the test case has its identifier, Repetition ID). Besides a list of all execution times of a test case, ALGator provides two pieces of information, the time of the first execution (Tfirst) and the time of the fastest execution (Tmin) of this test case. The first execution is usually much slower than other executions - as we will see in the following the Tfirst time can even be twice as big as the Tmin time. This behavior is more noticeable in the java environment since the JVM needs to warm up before it can operate at full speed [9].

To measure the time in Java we can only use the wall clock (Java does not provide any processor usage information). To minimize the unreliability of the measured time (which is due to the fact that the process may spend time waiting for I/O or for other processes that are also using the CPU) we use a "clean" computer which is dedicated only to the execution of the algorithms. Besides that, we usually take the Tmin time as reference data, since this is a time in which the computer is capable of solving the problem (the number of disturbing factors is minimal). For the algorithms implemented in the C programming language, we use the CPU time obtained by the `clock()` function (which returns the number of clock ticks used by the process). By calling this function before and after the algo-

```
void bubbleJ(int data[]) {
  int n = data.length;
  for (int i=0; i<n-1; i++) {
    for (int j=0; j<n-i-1; j++) {
      int a = data[j];
      int b = data[j+1];
      if (a > b) {
        data[j]   = b;
        data[j+1] = a;
      }
    }
  }
}
```

(a) BubbleJ (Java)

```
void bubbleC(int data[], int n) {

  for (int i=0; i<n-1; i++) {
    for (int j=0; j<n-i-1; j++){
      int a = data[j];
      int b = data[j+1];
      if (a > b) {
        data[j]   = b;
        data[j+1] = a;
      }
    }
  }
}
```

(b) BubbleC (C)

```
;esi=@data, ecx=n
    xor eax, eax
loop1:
    cmp eax, ecx
    je  lend
    mov ebx, 1
    sub ecx, eax
loop2:
    cmp ebx, ecx
    je  eloop1
    mov edx, [esi + 4*ebx-4]
    mov edi, [esi + 4*ebx]
    cmp edx, edi
    jle eloop2
    mov [esi + 4*ebx-4],edi
    mov [esi + 4*ebx],  edx
eloop2:
    inc ebx
    jmp loop2
eloop1:
    add ecx, eax
    inc eax
    jmp loop1
lend:
```

(c) BubbleA (assembler x86)

Figure 1: The tree implementations of the BubbleSort algorithm

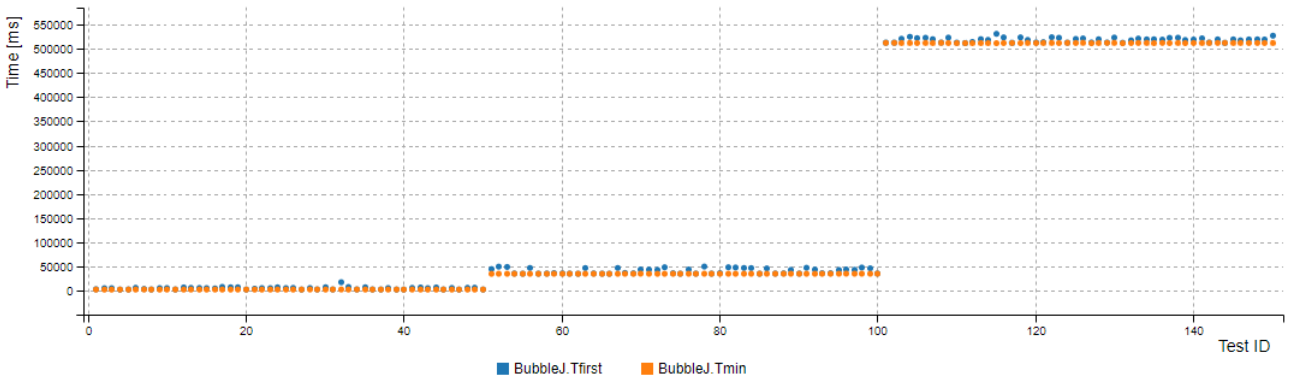


Figure 2: The Tmin and Tfirst measured times for `BubbleJ` implementation on inputs of size n=500, n=5000 and n=20000.

rithm execution and subtracting the returned values we get the total amount of time a process has actively used a CPU. The time measured this way is a much more reliable and accurate quality indicator.

3 The meaning of test repetition

In our first experiment, we would like to find out the meaning of several repetitions of a given test case execution. For this, we used a test set consisting of three groups of test cases: in each group, there are 50 identical tests of sizes 500, 5000, and 20000. All the input arrays in these test cases were ordered in inverse order (to ensure an identical number of operations during the sorting process). We executed each test case 50 times.

The graph in Fig. 2 depicts the times Tfirst and Tmin for all 150 tests. The Tfirst times (blue dots) in this graph are a little bit bigger than the Tmin times.

By analyzing the results we noticed that the (absolute) difference between Tfirst and Tmin is approximately the same for all three groups of test cases. The relative difference is therefore smaller for bigger measured times. We can conclude that the measurement of both Tfirst and Tmin is important for small inputs and that the importance of distinguishing between Tfirst and Tmin decreases with increasing input size. Measurements have shown that something similar to Java's "Tfirst phenomenon" also happens with C, except that in this case "warming up the machine" adds significantly less to the overall time complexity, so the differences in speed between Tfirst and Tmin are noticeable only in experiments that take very little time. From Table 1, which shows the relationship between the average first and the minimum execution time of a test case, $f = \frac{\overline{T}_{first}}{T_{min}}$ it can be seen that for small n the ratio is similar in both implementations, but for larger n the difference between Tfirst and Tmin is almost negligible for the `BubbleC3`, while for the `BubbleJ` the value decreases significantly more slowly. At $n = 20000$ the difference is still more than 5%.

The difference in measured times of multiple executions of the `BubbleJ` and `BubbleC3` implementations is depicted in Fig. 3. Here we used 50 inversely ordered arrays of size 5000, each test case was repeated 50 times. On the graph, the time of the first execution of the test case is shown in gray (Tfirst, Repetition ID=0), the first 20 repetitions are shown in orange and the next 30 in red.

With `BubbleJ`, we see that the first times (Tfirst) deviate considerably from the other measured times; the Tfirst times are somewhere between 18k and 22k, and the other times are much smaller (between 12k and 14k), which corresponds to the factor of 1.4 from Table 1. Other measured times on this graph do not show much fluctuation, as the scale of the display is reduced due to the large Tfirst times; we see that some Tfirst times are almost 100% larger than the smallest measured times. With `BubbleC3`, all times are quite similar to each other; the graph shows some variations, but everything is between 14.6k and 15.6k; the differences between measured times are relatively small (approx. 6%).

In conclusion: is it important to repeat the algorithm execution for several times to find the minimum time? As the measurements show, the answer depends on the size of the input - the smaller the input, the more measurements are unreliable, so we need to take more measurements to get a good result.

Bar charts in Fig. 4 depict the proportion of measurements that differ from the smallest measurement by the given percentage range. The measurements on small inputs for the `BubbleJ` vary a lot. More than 36% of all measurements differ from the minimal time of more than 10%. For the `BubbleC3` on the other hand only 17% of the measurements are that bad. When increasing the size of the input the results for both algorithms improve. For n=20000, for example, more than 73% (98%) of measurements differ from the minimal measurements for less than 1% for `BubbleJ`

N	*BubbleJ*	*BubbleC3*
500	1.98	1.80
5000	1.40	1.02
20000	1.05	1.01

Table 1: The ratio $f = \frac{\overline{T}_{first}}{T_{min}}$ between the average of the first and the minimal measured times

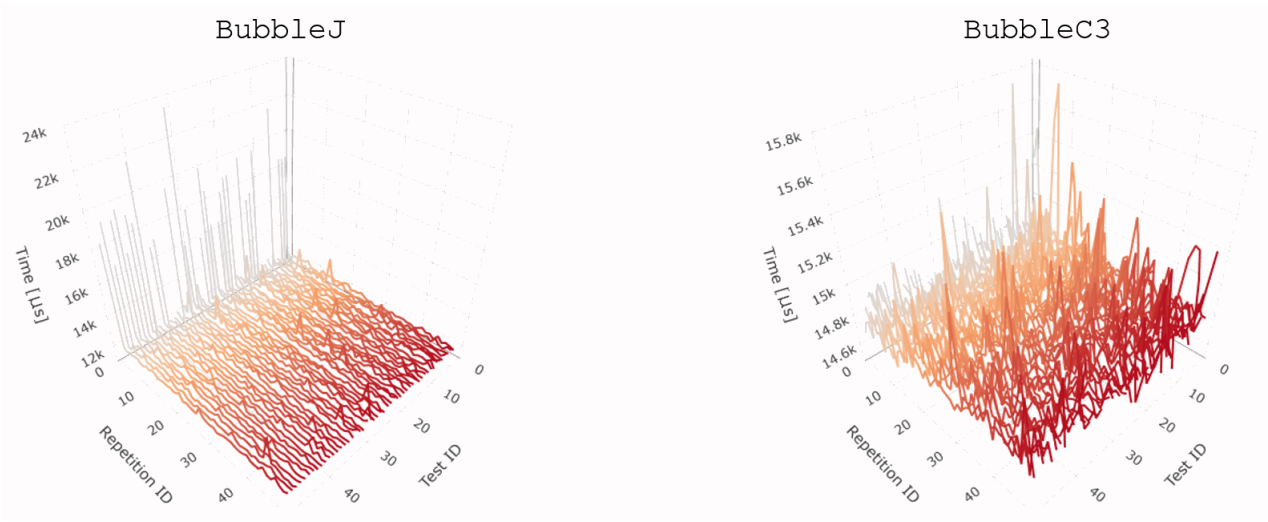


Figure 3: Times of execution of 50 identical test cases (50 repetitions of each test case) with `BubbleJ` and `BubbleC3`

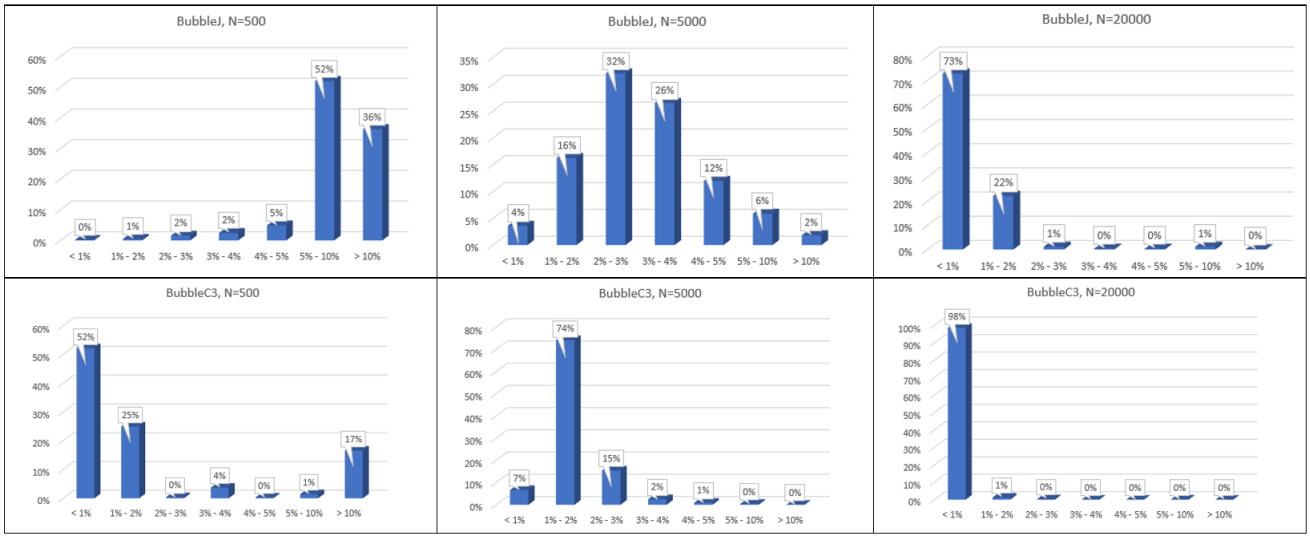


Figure 4: The proportion of measurements that differ from the smallest measurement by the given percentage range.

(`BubbleC3`) implementation.

The relative standard deviations of all measured times for `BubbleJ` are 21%, 7%, and 1% for n=500, 5000, and 20.000 respectively. This confirms the claim that as the size of the input increases, the importance of multiple tests decreases. Since the relative standard deviations are even smaller for `BubbleC3` (namely 15%, 1%, 0.24%), the importance of a large number of measurements is even smaller here.

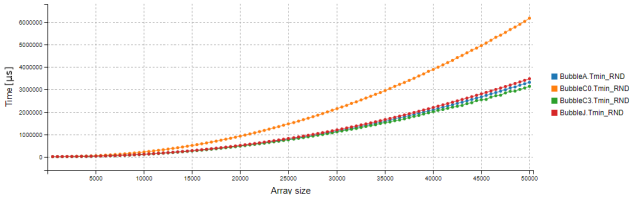


Figure 5: Tmin, RND data, $n = 500, ..., 50000$

4 The impact of the programming language

We compared the times of execution of four implementations (`BubbleJ`, `BubbleA`, `BubbleC0` and `BubbleC3`) on randomly ordered sequences (RND) of length 500 to 50000 (step 500). Each test was executed 30 times. Fig. 5 shows the minimum measured times Tmin of all four algorithms.

We expected the `BubbleC3` to be the best, which was also proven with the measurements. The difference between `BubbleC0` and `BubbleC3` is somewhat surprising. Since BubbleSort is a simple algorithm, one would expect that the speedup resulting from the optimization would not be that

great. But this is not the case, the difference is almost 2 times for large n. An interesting observation is that the factor of 2 appears to approximate the ratio between the sizes of machine code produced by optimized and non-optimized compilation. Specifically, the former contains 28 machine code instructions, while the latter contains 53.

The relationship between `BubbleJ` and `BubbleA` is interesting. In a battle between fast implementations, Java turned out to be the slowest, although the differences in speed are not so great. Fitting all measurements with quadratic functions results in the following:

`BubbleC0`: $T_{min}(n) = 2.438n^2\ \mu s$
`BubbleJ`: $T_{min}(n) = 1.372n^2\ \mu s$
`BubbleA`: $T_{min}(n) = 1.311n^2\ \mu s$
`BubbleC3`: $T_{min}(n) = 1.246n^2\ \mu s$

The ratio between the best (`BubbleC3`) and the worst (`BubbleC0`) implementation is 1 : 1.956, which we also noticed from the graph. More interesting is the ratio between the optimized C3 and Java implementation: `BubbleC3` : `BubbleJ` = 1 : 1,101. This means that for sorting random sequences Java is 10% slower than C. To find out, how good this conclusion is, let us calculate and depict the relative error

$$Error = \frac{|BubbleC3.T_{min} - 1.1 * BubbleJ.T_{min}|}{BubbleJ.T_{min}}$$

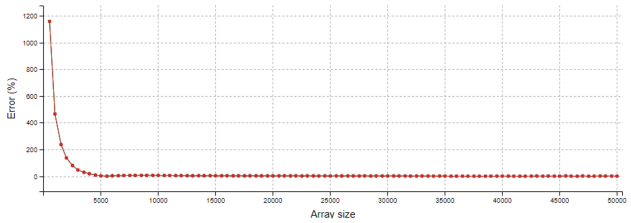


Figure 6: Relative error of estimating `BubbleJ`.Tmin with 1.1*`BubbleC3`.Tmin

Fig. 6 shows that for small inputs ($n < 5000$) the error is very big (as big as 1200%), but for larger inputs ($n > 10000$) the error is always less than 5% and it seems that it decreases when n increases.

5 The impact of the input data type

The input for a sorting algorithm is not always a randomly ordered array - sometimes the input data is already partially sorted. To find out if the (partial) pre-ordering of data impacts the execution time we used two special cases of the input data type - already sorted (SOR) and inversely sorted (INV) data. The results for both types show similar trends. Fig. 7 depicts the results on inversely ordered sequences (INV) of length 500 to 50,000 (step 500). The quality ranking of algorithms when sorting INV data changes compared to the ranking on RND data (Figures 5 and 7). While `BubbleC0` remains the worst implementation, in the first place there is a swap - `BubbleC3` gives way to `BubbleA` and `BubbleJ`. Something similar happens with the sorted (SOR) data. This change in ranking is hard to explain, but according to the research results presented in the following, we could speculate that the code generated by JVM is less suitable for branch prediction: with INV and SOR data the branch predictor is always correct, which could reflect better performance. Anyway, the results unequivocally show that the type of input has a great impact on the quality of implementation. While with random data `BubbleC3` implementation was faster than `BubbleJ`, for inversely ordered and already ordered data the Java implementation is the fastest.

The importance of data type is demonstrated by the results of the following experiment in which we ran the same algorithm (BubbleJ) on three different types of data: randomly ordered (RND), reverse-ordered (INV), and already sorted (SOR) data. With this experiment, we compared the number of swaps needed to sort an array with the time of execution and we discovered a strange behavior that can be explained only by the presence of the processor's branch predictor. The BubbleJ algorithm is a simple algorithm composed of three parts: loop administration, data comparison, and data exchange (data swap). Loop administration takes the same amount of time regardless of the input data type. Likewise, the input data type does not affect the number of comparisons performed by the algorithm, which is always precisely $n * (n-1)/2$. The input data type only affects the number of swaps performed, which is exactly $n * (n-1)/2$ in the case of INV data, 0 in the case of SOR data, and approximately $n^2/2$ in the case of RND data. The exact number of swaps in our test is shown in the left graph in Figure 8. Since the number of swaps is the only variable quantity during algorithm execution (the number of all the other operations is the same for all input data types), one would expect that graphs depicting the experimental time complexity for these three data types would be similar, but this is not the case. The right graph in Figure 8 shows that running the algorithm on randomly sorted data is much slower than running it on reverse-ordered data, even though the algorithm in the former case performs fewer operations than in the latter case. The only reasonable explanation for this phenomenon is in the influence of the processor's branch prediction mechanism, which optimizes data preparation for the processor and enables faster execution. This mechanism is particularly effective when successfully predicting the future, which it apparently does

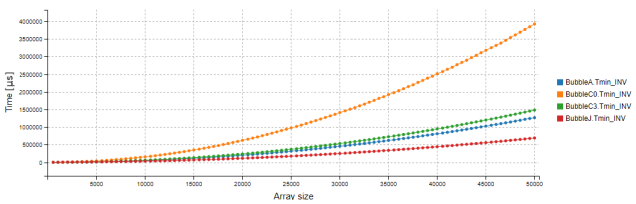


Figure 7: Tmin, INV data, $n = 500, ..., 50000$

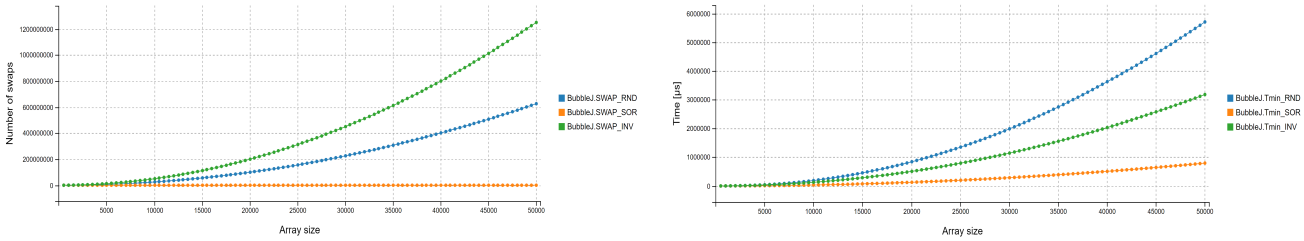


Figure 8: The number of swaps and the time of execution of BubbleJ algorithm on three different types of data: randomly sorted, already sorted and inversely sorted data.

well in the case of reverse-ordered data - because a swap follows each comparison, the branch predictor makes fewer mistakes than when swaps occur only occasionally. Properly prepared data makes the processor's work easier, resulting in a shorter total execution time, even though more operations are performed. This result clearly shows that the execution time of an algorithm is also influenced by factors that are not usually considered in theoretical analysis, which leads to significantly different results between theoretically predicted and empirically measured times.

6 Conclusions

The findings of this research paper demonstrate that the performance of algorithms is affected by numerous factors. Despite our efforts to maintain a controlled environment, we observed variations in our measurements. These deviations were particularly noticeable in Java, where the measurement of time is more sensitive to environmental influences than in C. We discovered that repeated execution of algorithms is particularly important for small inputs. Furthermore, we compared the performance of algorithms implemented in different programming languages. Our findings revealed that the difference between Java and C is not significant and that it depends on the type of input data. For randomly sorted arrays, the C implementation outperformed Java, while for inversely ordered and already sorted data, Java was superior.

In the future, we could apply similar methods to investigate other problems and determine if these results can be generalized. Additionally, we could explore the use of other popular programming languages such as Python, and examine in detail the real impact of the branch predictor on the final results. By conducting further research in this field, we can gain a better understanding of the factors that impact algorithm performance, and ultimately improve the efficiency and effectiveness of computer programs.

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[2] T. Dobravec. Algator — an automatic algorithm evaluation system. *Advances in Computers*, 116(1):65–131, 2019. https://doi.org/10.1016/bs.adcom.2019.07.002.

[3] T. Dobravec. Exact time measuring challenges. In *Proceedings of the 25th International Multiconference Information Society, IS MATCOS*, volume I, pages 21–24, Koper, 13-14 October 2022.

[4] M. Fernández. *Models of Computation, An Introduction to Computability Theory*. Springer, 2009. https://doi.org/10.1007/978-1-84882-434-8.

[5] D. Johnson. A theoretician's guide to the experimental analysis of algorithms. 12 2001. https://doi.org/10.1090/dimacs/059/11.

[6] R. Kumar. *Instruction Level Parallelism: Branch Prediction and Optimization*. LAP LAMBERT Academic Publishing, 2012.

[7] C. C. McGeoch. Experimental methods for algorithm analysis. *Encyclopedia of Algorithms*, 2008. https://doi.org/10.1007/978-0-387-30162-4_135.

[8] B. Moret. Towards a discipline of experimental algorithmics. *Monograph in Discrete Mathematics and Theoretical Computer Science*, 2002. https://doi.org/10.1090/dimacs/059/10.

[9] M. Price. Hot code is faster code - addressing jvm warm-up. *QCon*, April 2016.

[10] B. Swathi. A comparative study and analysis on the performance of the algorithms. *International Journal of Computer Science and Mobile Computing*, 5(1):91–95, Januar 2016. ijcsmc.com/docs/papers/January2016/V5I1201621.pdf.

[11] M. Tedre and N. Moisseinen. Experiments in computing: A survey. *The Scientific World Journal*, (1):1–11, 2014. https://doi.org/10.1155/2014/549398.

https://doi.org/10.31449/inf.v48i4.4768 Informatica **48** (2024) 549–566 **549**

Surrogate Estimators For Collaborative Decision

Fatiha Bendali, Alejandro Olivas Gonzales, Alain Quilliot, Hélène Toussaint
Limos CNRS 6158, Labex IMOBS3, Clermont-Ferrand, France
E-mail: alain.quilliot@uca.fr

Keywords: Scheduling, combinatorial optimization, multi-level optimization, machine learning, energy management

Received: April 11, 2023

We deal here with job scheduling under the assumption that performing a job requires the production of encapsulated renewable and non-renewable resources. For the sake of understanding, we rely here on a case study related to energy production by a photo-voltaic platform. Handling it means synchronizing resource production and consumption in order to optimize both production costs and some scheduling criterion, and induce the setting of a complex bi-level model. Moreover, this applicative context makes appear that job scheduling and resource production most often depend on distinct players, provided with their own agenda and access to information. Adopting here the point of view of one specific player, namely the job scheduler, leads us to set a model that shortcuts the production level with the help of surrogate estimators. Those estimators involve flexible pricing mechanisms and machine learning devices. According to this, we first perform a structural analysis of our model, before designing and testing several algorithms that implement this surrogate component based approach.

Povzetek: Poročilo obravnava problem načrtovanja nalog z uporabo obnovljivih in neobnovljivih virov energije, kjer se osredotoča na sinhronizacijo proizvodnje in porabe energije. Predlaga uporabo nadomestnih ocen za obvladovanje kompleksnih dvo-nivojskih odločitev v sodelovalnih okoljih, pri čemer vključuje tudi strojno učenje za optimizacijo stroškov.

1 Introduction

The notion of multi-level decisional model (see [16], [18]) most often derives from contexts involving several players, independent from each other or tied together by some hierarchical or collaborative link, who share the decision with respect to some system. Solving such a model aims either at providing a best scenario if all the players operate under a common authority (centralized paradigm), or (collaborative paradigm) at helping them into the search for a compromise.

Most contributions address multi-level models according to the centralized paradigm, while assuming the existence of a unique decider provided with full information. Handling a model set this way is a difficult task (see [10], [11]). It usually involves complex decision sub-models of very different types. Standard approaches rely on decomposition schemes, hierarchical (Benders decomposition, Stackelberg Equilibrium,…) or horizontal (Lagrangean relaxation) (see [2], [26]). In both cases a major difficulty derives from the sensitivity issue, which means the way one may retrieve information from the different levels in order to make them interact. In case the decision problem involves temporal constraints about jobs to be processed, another difficulty comes from resulting synchronization constraints (see [9]), which require the different players to meet in order to exchange resources or informations.

Yet in practice, it may be utopian, for both technical and economical reasons, to assume that all players will agree on a common agenda and on the share of information. *Game Theory*, mainly *Cooperative Game* (see [7], [8], [21],[33]), provides us with a useful tool for the anticipation of the behavior of the players and for the distribution of the costs among them. But it does not help a specific player in making its own decision. So, when the focus is on such a specific player, one must find a way to bypass the levels related to the other players and replace, under incomplete information, the criteria and costs related to those players by surrogate estimators.

Present contribution reflects this concern. If we refer to the Frascati Manual of the European Commission, it refers to Section I.2 (Informatics and Information Sciences) while involving developments related to Combinatorial Optimization and Operations Research. It refers to fundamental research in the sense that collaborative decision and complex multi-level models are generic issues, and it has also to see with Applied Research since, in order to make things easier to understand, we start from an applicative context. This context is related to researches conducted about intelligent vehicles inside the IMOBS3 (*Innovative Mobility Services, Systems and Structures*) Labex in Clermont-Ferrand, and to a partnership between LIMOS Laboratory and the national PGMO (*Gaspard Monge for Optimization*) program promoted by power company EDF

(*Electricity De France*). It involves the joint management of energy by a local photovoltaic (PV-Plant) platform and by a consortium of users (industrial players, services providers, …) relying on this energy in order to perform jobs according to their own purposes. The fact is that market deregulation and emergent technologies currently induce the rise of local renewable energy producers (factories, farms, householders, ...) who simultaneously remain consumers. Those new players make *self-consumption* become an issue (see [5], [22], [30], [36], [44]) whose key operational feature is the need for synchronization between time-dependent resource production and its consumption (see [12], [14], [23], [25]).

So we consider here on one side, a production manager who runs a PV-Plant (Photovoltaic Plant), that not only produces energy and distributes it among end-users (jobs) but also buys and sells energy on the market. On the other side, one or several *job schedulers* are in charge of performing specific jobs. Both meet in order to perform *recharging transactions*: in order to avoid jobs to waste time while waiting for recharge, the PV-Plant relies on a set of batteries and implement a *swapping* policy (see [4], [40], [41]), so that the job schedulers only need to switch from a battery to another one. We suppose that this *plug out/in* operation is instantaneous. Limited storage and recharge capacities impose both players to carefully synchronize the time-dependent energy production and its consumption. This requirement makes resulting bi-level decision problem complex, even under the centralized paradigm. Many searchers recently showed interest into the decisional problems raised by the management of renewable energy. They most often focused either on production scheduling (see [1], [13], [15]) or on the issues related to consumption (see [3], [17],[19], [27], [31]). But very few dealt with the interaction between both (see [5], [6], [23], [34], [36], [40]) and they most often did it while adopting the point of view of a unique decider provided with full information. Our goal is to study here this interaction and the way players collaborate under incomplete information (see [12], [29], [34], [35]), in order to derive heuristic algorithms reflecting the point of view and the knowledge of a specific player. Though part of the difficulty of our specific application is due to uncertainty, we suppose, for the sake of simplicity, that our system behaves in a deterministic way.

Starting from this applicative context, we set a bi-level **PVSync** model. This model is a centralized macroscopic model, which relies on simplifications with respect to the behaviors of respectively the batteries and the PV-Plant (linearization of the charge and discharge processes for the batteries (see [37], [38], [39]), deterministic power prediction for the PV-Plant (see [44]). Those simplifications are justified not only by our wish to get a tractable model, but also by the fact that we intend adopting the point of view of the job scheduler and designing heuristic algorithms that reflect this point of view. We notice that this **PVSync** model may be viewed as a new variant of the well-known **RCPSP**: *Resource Project Scheduling Problem*(see [32], [24]). **RCPSP** is about scheduling jobs under temporal and resource constraints, and our model introduces non-renewable resources (energy) encapsulated into renewable ones (the batteries).

We first study **PVSync** according to the centralized paradigm. We cast it into the MILP: *Mixed Integer Linear Programming* framework, and analyze the structure of its main components. Our main purpose being the design of heuristics reflecting the point of view of the job scheduler, we introduce a *projection* mechanism that projects the constraints related to the batteries, that the job scheduler does not control, into a surrogate *Idle Battery* constraint that he may easily handle. Next, we address the collaborative issue. We start doing it while supposing that the production manager acts as a mediator, and cast our problem into the *Cooperative Game* framework. We keep on while adopting the point of view of the job scheduler, who implicitly endorses the role of the master of the game. Our goal becomes the design of *job scheduler* oriented heuristic algorithms *SurrPVCost* that handle **PVSync** according to this restricted point of view, while short-cutting the part of the process involving the PV-Plant. We use a *surrogate* formulation of the costs together with *surrogate* constraints that aims at making the hidden production level remain feasible. The *surrogate* constraints are the *Idle Battery* projected constraints that we just mentioned above, augmented with additional constraints. As for the *surrogate* formulation of the costs, we try two approaches: the first one is based upon a pricing mechanism; the second involves machine learning and convolutional neural networks (see [28], [43]). Both approaches are well-fitted to the management of the non-deterministic case.

So the paper is organized as follows. In Section 2, we introduce the **PVSync** problem. We set it according to the centralized MILP (*Mixed Integer Linear Programming*) framework (Section 2.2), and discuss different formulations (Section 2.3). In order to help the reader in identify the scope of the paper we fix the limitations of this research in Section 2.4.In Section 3, we analyze the structure of the main components of this model and the properties of the *Idle Battery* constraint. We start discussing in Section 4 the cooperative issue and first cast in Section 4.1 our problem into the *Cooperative Game* framework. Next we introduce in Section 4.2 a generic *collaborative* algorithmic scheme for the handling of **PVSync** while using surrogate estimators. We describe in Section 4.3.1 the surrogate constraints that we derive from the projection scheme of Section 3, and propose in sections 4.3.2 and 4.3.3 2 estimators of the production costs: the first one relies on a parametric pricing mechanism while the second one involves a convolutional neural network. We describe in Section 4.4 the way those estimators may

be used in order to drive a *job scheduler* oriented heuristic algorithm *SurrPVCost*, and devote Section 5 to numerical experiments.

2 The PVSync problem

This section is devoted to the description of our case study, related to energy management, which will be the starting point for coming developments. As told in the introduction, this case study derives from a partnership between the LIMOS Laboratory and the national PGMO (*Gaspard Monge for Optimization*) promoted by power producer EDF. It provides us with a medium for the implementation and the tests of algorithmic proposals related to a more fundamental issue, namely the management of complex multi-level decision models in both centralized and collaborative contexts. According to this, our main contribution will be about the way we may endorse the point of view of a given specific player, set a model reflecting its agenda and access to information, that consequently reduces the behavior of the other players to what we call *surrogate* constraints and criteria, and accordingly design heuristic algorithms.

While setting a model for this case study, we shall notice that it extends one of the most fundamental scheduling problem, the **RCPSP**: *Resource Constrained Scheduling Problem*.

The Job Scheduler Side: We consider a set of *jobs* $\mathbf{J} = \{1, \ldots, J\}$ to be performed exactly once within a time horizon $[0, N]$ divided into unit length periods. We denote by i the period $[i-1, i]$. Any job j requires t_j periods and is constrained by a time window $\{Min_j, \ldots, Max_j\}$: j must start no sooner than period Min_j and end at period no later than Max_j. Some pair of jobs (j_1, j_2) are tied together by some precedence relations $j_1 << j_2$ whose meaning is that j_1 must be finished when j_2 starts. We do not allow *Preemption*: Once a job starts, it cannot be interrupted.

The Resources : Every job j requires some amount of (electric) energy e_j. This energy is stored inside a set $\mathbf{K} = \{1, \ldots, K\}$ of identical batteries, that are assigned to the jobs at the time when they start. Every job requires exactly one battery and a battery cannot be simultaneously assigned to several jobs. It comes that no more than K jobs may be running at the same time. A battery must be periodically recharged in order to run as many jobs as possible, and *recharge* takes place when the battery is *idle* and does not run any job. The *storage* capacity of a battery is denoted by C: The amount of energy stored inside a battery cannot exceed C. The *recharge* capacity of a battery is denoted by C^R: The amount of energy which may be loaded into a battery during 1 period cannot exceed C^R. The initial load of battery $k \in \{1, \ldots, K\}$ is denoted by H_k^{Init}. For technical reasons, assigning a battery to a job j takes place at the junction between 2 periods, that means at some time $i, i = 0, \ldots, N-1$. Such a *swapping* transaction is instantaneous. The battery remains *active* during the time when j is running, and cannot be recharged during this time. We introduce the following notations:

- $\hat{e} = \sum_j e_j$; $\hat{t} = \sum_j t_j$; $E^{Mean} = \frac{\hat{e}}{\hat{t}}$;
- For any j, $e_j^{Mean} = \frac{e_j}{t_j}$.

Remark 1: We proceed here in a way that is standard with respect to this kind of problems (see [3], [19], [27]) and that consists in simplifying the physics of both the batteries and the PV-Plant. We implicitly refer here to the same kind of Ion/Lithium batteries that may be embedded into electric vehicles, with storage capacities between 20 KWh and 100 KWh, and recharge capacities between 10 KW and 100 KW. However, we do as if that the charge and discharge processes of those batteries were linear, that may be considered as true as long as the current charge of the battery remains located inside some critical interval (see [37], [38]). This implicitly means that capacity C is not the real capacity of the battery, rather the size of the interval inside which the charge and discharge processes may considered as quasi-linear. Also, we do not allow here any choice choice between different recharge capacities involving distinct economic costs.

The PV-Plant Side: In order to implement a *self-consumption* policy we rely on a *PV-Plant*, that means on a photovoltaic facility. This PV-Plant assigns the batteries to the jobs according to some swapping policy and produces its own energy that it distributes among the currently idle batteries or that it sells to the market. In case this self-produced energy is not enough, the PV-Plant can also buy energy to the market. Energy stored inside the batteries can be neither transferred to another battery nor sold. We denote by R_i the expected production of the PV-Plant at period i, by P_i the energy unit purchase price at period i, and by S_i the energy unit sale price. Of course we have, for any i, $P_i \geq S_i$. An example of cost and production rates is provided by Figure 1. We set, for any period i:

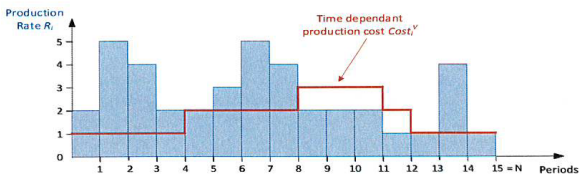


Figure 1: Time dependent production rates and costs for the micro-plant

- P^{Mean} = mean value $P_i, i = 1, \ldots, N$;

– S^{Mean} = mean value $S_i, i = 1, \ldots, N$;

– R^{Mean} = mean value $R_i, i = 1, \ldots, N$.

Figure 2 describes the way the PV-Plant and the jobs interact in the case when the jobs are tours performed by electric vehicles, that start and end into the PV-Plant and make the vehicles visit a set of customers. It shows the swapping policy that makes some *active* batteries be embedded into the vehicles, while the other *idle* batteries remain available for recharge.

Remark 1-Bis: We bypass the physics of the PV-Plant (see [39], [44]) and consider its behavior as deterministic, following the way standard software like PVLib Matlab ([44]) proceed.

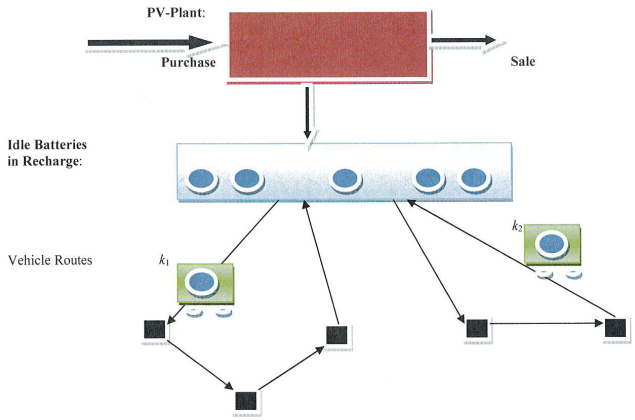


Figure 2: PV-Plant, batteries and *vehicle routing* jobs

Then we may set the **PVSync** problem:

PVSync: {Simultaneously schedule the jobs and the PV-Plant, in such a way that:

– Every job j is run once without any interruption (*Non-Preemption*) within its time window. Precedence constraints $j_1 << j_2$ are satisfied. (C1)

– Every time a job j is performed, it is provided with a battery $k(j)$ loaded with at least e_j energy. (C2)

– The global energy load of the batteries at the end of period N must be at least equal to $\sum_k H_k^{Init}$. (C3)

– Some hybrid cost $\alpha.SchCost + PVCost$ is minimized. $PVCost$ is the *self-consumption* cost, equal to the difference between the energy purchase costs and the profits derived from the sales. *Scheduling* cost $SchCost$ is the sum of the completion times of the jobs.}

Remark 2: PVSync and the RCPSP problem: The **PVSync** problem may be viewed as new variant of the well-known **RCPSP**: *Resource Constrained Scheduling Problem* (see [32], [24] for standard RCPSP), that is itself an extension of the *Multi-Processor Scheduling* problem. Jobs depend here on *encapsulated* resources, which are renewable batteries and non-renewable energy embedded into the batteries. The first ones act as *containers* for the second ones. Setting this extension in a formal way would imply introducing a lot of notations, without any fundamental impact on the methods. So, instead of doing it, we keep on with the **PVSync** setting and terminology, while referring to the batteries as renewable container resources and to the energy as non renewable content resource.

Remark 3 : We may set **PVSync** as a bi-level model by distinguishing the respective roles of the job scheduler and the PV-Plant :

Bi-level Reformulation of PVSync: {Schedule the jobs in such a way :

– (C1) is satisfied.

– An hybrid cost $\alpha \cdot SchCost + PVCost$ is minimized, $PVCost$ being the optimal value of the following **PVPlant** sub-problem :

PV-Plant sub-problem : {Schedule the sale, purchase and distributions operations, and assign the batteries to the jobs in such a way that (C2) is satisfied and that the difference between the energy purchase costs and the profits derived from the sales is minimal.}}

Remark 4: One might clearly make the PV-Plant side more complex, by broadening the scope of feasible sale, purchase and distribution operations, by getting closer to the physics of storage and recharge/discharge processes, by involving Heterogeneous batteries, with distinct storage and recharge capacities, and by allowing distinct recharge capacities, each recharge capacity involving its own specific costs.

Our Goal and Contribution : As a new bi-level extension of the RCPSP problem, the **PVSync** problem is interesting by itself. Yet, our purpose here is not the design of exact algorithms for this problem, rather the study of the way one might efficiently handle it if the PV-Plant processes become more complex (see Remark 4) or in the case of a collaborative context, that would impose us to focus on the job scheduler point of view and cope with restrictions on the information related to the behavior of the PV-Plant. This means that our main contribution here is the design of heuristic algorithms that tends to emulate the decentralized point of view of a specific player, namely the job scheduler, possibly provided with partial information. In order to achieve it, we first cast **PVSync** into the MILP framework (Section 2.2), in order to get an unambiguous setting and to provide ourselves with reference results, and discuss

some variants (Section 2.3). Next we perform some structural analysis (Section 3) with focus on the introduction of a mechanism that projects the constraints related to the batteries (that the job scheduler does not control) into surrogate *Idle Battery* constraints that he may easily handle. The introduction of this projection mechanism opens the way (Sections 4 and 5) to the design and test of *job scheduler* oriented heuristic algorithms *SurrPVCost* for the handling of **PVSync**, that reflect the incomplete point of view of the job scheduler. These algorithms rely on the management of surrogate constraints, (the *Idle Battery* constraints augmented with some additional constraints), and of surrogate objective functions, that express the approximation that the job scheduler may get of the part of the costs that it does not fully control. Though we stick here to our case study, this approach is generic and might be applied to other multi-level/multi-player decision problems.

2.1 An example

Let us suppose that $J = \{A, B, C, D, E\}$, with respective durations $t_j = 2, 1, 2, 3, 1$ and energy requirements $e_j = 5, 5, 4, 9, 4$. Jobs A and B must be run between periods 3 and 5, job C between periods 5 and 8, job D between periods 2 and 7, and job E between periods 7 and 10.
We are provided with 2 identical batteries k_1 and k_2, with initial loads respectively equal to 7 and 6. We have: $C = 12$, $C^R = 3$. The time space is divided into 10 periods and time versus money coefficient α is equal to 2. Production data come as in table 1:

Table 1: Prices and production coefficients

i	1	2	3	4	5	6	7	8	9	10
P_i	2	3	7	7	3	2	6	7	4	2
S_i	1	2	4	4	1	1	3	3	2	1
R_i	4	4	3	5	2	6	4	4	4	5

Then we get (Fig. 3) a feasible **PVSync** solution:

- Battery k_2 consecutively runs jobs A and B between periods 3 and 5 and comes back to the PV-Plant at the end of period 5. It reloads until period 7, runs job E and comes back to the PV-Plant.
- Battery k_1 runs job D at period 2 and comes back to the PV-Plant at the end of period 4. Then it reloads before running job C at period 8.

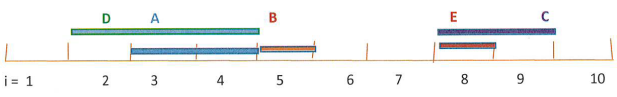


Figure 3: A feasible schedule of the jobs

For every period i, energy amounts respectively bought, sold and distributed to the batteries are given by table 2.

Table 2: Scheduling the production

i	1	2	3	4	5	6	7	8	9	10
Bought	1	0	0	0	1	0	0	0	0	1
Sold	0	1	3	5	0	0	1	4	2	0
To k_1	2	*	*	*	3	3	*	*	3	*
To k_2	3	3	*	*	*	3 *	*	2	3	*

2.2 A MILP model

The purpose of this section is to cast the **PVSync** problem into the MILP framework. This will provide us with a non ambiguous formal setting of this problem, that we will refer to all along the paper, as well as with reference results for numerical experiments. Notice that it would have been possible to choose another framework, like for instance the SAT framework (while using the library Google's CP-SAT, part of the OR-Tools Library) or the Constraint Programming framework (while relying on the IBM CPO software). But the fact is that the MILP framework is a kind of standard in Operations Research and Combinatorial Optimization, and is also widely used in the Industry, specifically in companies involved in Telecommunications, Transportation and Energy Production.

Since **PVSync** may be viewed as a **RCPSP** extension, we adapt the standard **RCPSP** MILP: *Mixed Integer Linear Program* formulation corresponding to the case when the time space is explicitly divided into unit-time periods. In such a case, **RCPSP_MILP** usually relies on a {0, 1}-valued vector $Z = (Z_{j,i}, i = 1, \ldots, N, j = 1, \ldots, J)$ which tells us at which period i a job j starts. In our case, we must also explicitly identify the batteries so that they may be assigned to the jobs. We link the jobs, the periods and the batteries through a {0, 1}-valued vector $X = (X_{j,k,i}, j = 1, \ldots, J, k = 1, \ldots, K, i = 1, \ldots, N)$: $X_{j,k,i} = 1$ means that battery k is assigned to job j, starting at period i.
On the other side, we represent the production activity of the PV-Plant by 3 non negative rational vectors $U = (U_i, i = 1, \ldots, N)$, $V = (V_i, i = 1, \ldots, N)$, $Q = (Q_{k,i}, k = 1, \ldots, K, i = 1, \ldots, N)$, with respective meanings:

- U_i means the energy purchase at period i,
- V_i represents the energy sale at period i,
- $Q_{k,i}$ represents the energy distributed to battery k at period i.

We link the PV-Plant, the jobs and the batteries through a last variable $W = (W_{k,i}, k = 1, \ldots, K, i = 0, \ldots, N) \geq 0$ with rational values: $W_{k,i}$ represents the energy inside battery k at the end of period i ($W_{k,0}$ means the initial load of battery k). Resulting model comes as follows:

PVSync_MILP Model:{Compute:

- $\{0, 1\}$-valued vector $Z = (Z_{j,i}, i = 1, \ldots, N, j = 1, \ldots, J)$: $Z_{j,i} = 1$ means that job j starts at period i.
- $\{0, 1\}$-valued vector $Y = (Y_{j,k}, k = 1, \ldots, K, j = 1, \ldots, J)$: $Y_{j,k} = 1$ means that battery k is assigned to job j.
- $\{0, 1\}$-valued vector $T = (T_{k,i}, k = 1, \ldots, K, i = 1, \ldots, N)$: $T_{k,i} = 1$ means that battery k is active at period i.
- $\{0, 1\}$-valued vector $X = (X_{j,k,i}, k = 1, \ldots, K, j = 1, \ldots, J, i = 1, \ldots, N)$: $X_{j,k,i} = 1$ means that battery k is assigned to job j, starting at period i.
- $U = (U_i, i = 1, \ldots, N) \geq 0$ with rational values: U_i means the energy purchase at period i.
- $V = (V_i, i = 1, \ldots, N) \geq 0$, rational: V_i means the energy sale at period i.
- $Q = (Q_{k,i}, i = 1, \ldots, N, k = 1, \ldots, K)$, rational: $Q_{k,i}$ means the energy distributed to battery k at period i.
- $W = (W_{k,i}, i = 0, \ldots, N, k = 1, \ldots, K) \geq 0$, rational: $W_{k,i}$ means the load of battery k at the end of period i.

Under the constraints:

- Minimize $\sum_i (P_i.U_i – S_i.V_i) + \alpha.(\sum_{j,i} i.Z_{j,i})$. (E1)
- For any i, j, $Z_{j,i} = \sum_k X_{j,k,i}$. (E2)
- For any k, j, $Y_{j,k} = \sum_i X_{j,k,i}$. (E3)
- For any battery k, period i_1, $T_{k,i_1} = \sum_{j,i=i_1-t_j+1,\ldots,i_1} X_{j,k,i}$. (E4)
- For any j, $\sum_k Y_{j,k} = 1 = \sum_i Z_{j,i} = 1$. (E5)
- For any j, any i s.t $(i < Min_j)$ or $(i > 1 + Max_j - t_j)$, $Z_{j,i} = 0$. (E6)
- For any k, i, $Q_{k,i} \leq C^R.(1 - T_{k,i})$. (E7)
- For any i, k, $W_{k,i} \leq C$. (E8)
- For any k, $W_{k,0} = H^{Init}_{k,0}$ and $W_{k,N} \geq H^{Init}_{k,0}$. (E9)
- For any $i \geq 1, k$, $W_{k,i} = W_{k,i-1} + Q_{k,i} – (\sum_j e_j.X_{j,k,i})$. (E10)
- For any i, $U_i + R_i = V_i + (\sum_k Q_{k,i})$. (E11)
- For any j_1, j_2 s.t. $j_1 << j_2$, $\sum_i i.Z_{j_1,i} + t_{j_1} \leq \sum_i i.Z_{j_2}$. (E12)}

Proposition 1: *Solving **PVSync_MILP** solves the **PVSync** problem.*

Proof: One easily turns any feasible solution of **PVSync** into a feasible solution (Z, Y, T, X, U, V, W, Q) of **PVSync_MILP** with same cost. Checking that every constraint (E1),…, (E12) is satisfied derives in a straightforward way from the meaning of those constraints: (E2) means that if job j starts at period i, then there is exactly one battery assigned to j. (E3) means that if battery k is assigned to job j, then there exists a unique period i such that j starts with battery k at period i. (E4) means that if battery k is active at period i_1, then there exists a unique job j which starts with battery k at i such that $i \leq i_1 \leq i + t_i - 1$. (E5) means that to any job j must be assigned exactly one battery and one starting period. (E6) means that any job j is run within its time window. (E7, E8) are capacity constraints: the load inside a battery at the end of a given period cannot exceed the *storage* capacity of the battery, and the amount of energy loaded into a battery during a given period cannot exceed the *recharge* capacity. (E9) expresses the initial and final requirements for the batteries. (E10, E11) are nothing more than balance equations, which distribute energy over the time between purchase, sale, storage and consumption by the jobs. (E12) means that if two jobs j_1, j_2 are such that j_1 precedes j_2, then the starting period for j_2 must be larger than the ending period of j_1. Conversely, the key point is that (E5) assigns exactly 1 battery k and 1 starting period i to every job, and that variables $X_{j,k,i}$ involved in (E2, E3, E4) allow us to characterize the periods when a battery is active or idle. This in turn enables us to control through (E7, E11) the amount of energy loaded into any battery at any period, as well as the *Non-Preemption* constraint. The rest of the proof comes in a straightforward way. **End-Proof**.

2.3 A short discussion: variants of the PVSync problem

In practice, production configurations may be more complex (see Remark 4), and we may be imposed to get closer to the physics of both the batteries and the PV-Plant. Let us mention here 3 possible variants, whose handling would provide us with a stronger motivation for the approaches that we are going to describe in Section 4:

First Variant: *Batteries may be used in order to store energy and sell it later.* According to this hypothesis, a battery k may receive energy at period i and sell it at period $i' > i$. The **PVSync_MILP** model must be updated through the introduction of an additional vector $V^S = (V^S_{k,i}, k = 1, \ldots, K, i = 1, \ldots, N)$, meaning the amount of energy sold at any period i by battery k.

Second Variant: *A fixed storage unit may be used either for later sale or battery feeding*. According to this hypothesis, such a *Buffer* battery $BUFF$, with storage capacity C^{BUFF}, initial load H^{BUFF}, recharge capacity C^{ReBUFF} and discharge capacity C^{DeBUFF}, induces the introduction of new variables into the model:

Q^{BUFF}_i = energy sent from the PV-Plant to the *Buffer* battery at period i.

V^{BUFF}_i = energy sold by the *Buffer* battery at period i.

$Q_{k,i}^{BUFF}$ = energy sent by the *Buffer* battery to battery k at period i.

W_i^{BUFF} = energy inside the *Buffer* battery at the end of period i.

Third Variant: *Periods units and time values do not coincide*.
Actually, the structure of the time space for the PV-Plant may not coincide with the structure of the time space for the jobs. More precisely, a period means a time interval during which the PV-Plant is stable (no change neither in the production rates nor in the purchase/sale prices). In most cases, the duration of a period is given as an integral number p, while the duration of a job may take any integral value. In such a case, representing the schedule of the jobs with a vector $Z = (Z_{j,i}, i = 1, \ldots, N, j = 1, \ldots, J)$ as in the previous section does not hold anymore and getting a **PVSync** MILP formulation requires the introduction of flow vectors representing the way the jobs exchange the resources.

2.4 Limitations of the research

The present paper involves a wide range of concepts and software tools. To better delineate the scope of the paper and remove any ambiguity, we will summarize its limitations here. This will help the reader distinguish between concepts mentioned as potential future research topics and methods used as tools, from the issues that are central to the paper.

- **The RCPSP is not central**: We mention in the introduction and in previous section 2.2 that **PVSync** may be viewed as an extension or a variant of the standard **RCPSP** problem. This extension is novel in that it links renewable and non-renewable resources through an encapsulation relation (the renewable batteries are containers for the non-renewable energy). In the future, it may be interesting to further study **RCPSP** models linking renewable and non-renewable resources and to design new exact algorithms or adapt existing ones. However, this is not the purpose of this paper, which is in no way focused on the **RCPSP**. We do not cast **PVSync** into the generic **RCPSP** framework involving jobs and resources, nor do we focus on designing exact algorithms for handling it (typically, contributions about **RCPSP** presuppose a centralized context and emphasize the design of fast-running exact algorithms [24]).

- **MILP models and solvers are tools, not the focus**: At the beginning of former section 2.2, we provide a MILP formulation of **PVSync**. The main purpose of this MILP model is to establish an unambiguous formulation of the **PVSync** problem and to generate benchmark results that can help evaluate heuristic algorithms designed from the restricted perspective of a specific player. We use the MILP framework primarily because we are familiar with the CPLEX Library. Consequently, our focus is not on designing sophisticated MILP settings for **PVSync** that involve valid cuts, facets, or specific decomposition schemes to facilitate the computation of exact solutions. In realistic contexts, the complexity of the production level (the PV-Plant) and the incompleteness of the information available to a target player would render such settings impractical. Instead, our goal is to design heuristic algorithms that are adapted to such complex contexts.

- **The same applies to *Machine Learning***: Sections 4 and 5 of the paper will involve neural networks used to provide the job scheduler with an approximation of the behavior of its production partner. Since the behavior of this PV-Plant player is related to solving a production planning problem, we use convolutional neural networks (CNNs) to handle instances of this planning problem with varying sizes while limiting the number of synaptic coefficients. However, *Machine Learning* will not be our primary focus: We shall rely on standard, ready-to-use open software (Keras TensorFlow) that we apply to a CNN with a fairly intuitive structure, and we train this CNN using versions of the stochastic gradient algorithm implemented within this software. The accuracy of the neural network is not the critical factor here; rather, it is its ability to guide our heuristic scheduling algorithms toward satisfactory solutions.

- **PV physics and technologies are only a hypothesis**: Our paper addresses a decision problem from the perspective of a specific target player who bypasses some levels of the complex underlying system by using surrogate estimators. When referring to the physics behind this system, related to batteries and charge/discharge processes, we remain at a macroscopic level, and our decision models require significant simplifications. As explained in both the introduction and at the beginning of this section, these simplifications involve assuming that the recharge and discharge processes are linear and that the behavior of the PV-Plant is deterministic, which is generally accurate as long as the battery load remains within a safe interval. Therefore, we do not concern ourselves with the full infrastructure of the PV-platform (such as inverters and trackers) or what occurs at the microscopic level. In other words, our contribution is not about photovoltaic production technology, nor is it about the physics and electronics involved in the production and storage of this renewable energy.

3 Structural analysis

We analyze in this section the structure of **PVSync** and its complexity, while adopting the standard centralized point of view (one decider, provided with all information). We de-

scribe the projection mechanism (*Idle Battery* constraints) that will allow us to bypass the battery level. We first introduce the following notations:

For any subset A of the job set **J**, we denote by **PVSync**(A) the restriction of **PVSync** to the jobs of A, and by $PVSync(A)$ its value.

For any schedule vector Z we denote by **Restrict-PVSync**(Z) the problem which derives from **PVSync** by fixing Z, and by $RestrictPVSync(Z)$ its value.

For any schedule vector Z and any period i, we denote by $n(Z,i)$ the number of jobs which are active at period i according to Z, and by $L(Z,i) = K–n(Z,i)$ the number of batteries which are idle at period i, that means which are available for recharge.

Given a schedule vector Z and a job j, we denote by $Start(Z,j)$ the starting period of j according to Z.

3.1 The different levels of PVSync and their complexity

PV-Sync is a multi-level problem. We may distinguish a scheduling level (variables Z), a battery level (variables Y) and a production one (variables U, V, Q). This last one is nothing but a linear program with rational variables.

The Scheduling Level
Restricting **PVSync** to its scheduling level means short-cutting the activity of the PV-plant, and only considering the jobs, provided with durations, time windows and precedence constraints, together with K identical batteries which lose their container status and behave as machines in the standard scheduling sense. In the case of variant 3 of former section 2 this restriction of **PVSync** to its scheduling level would contain the multi-machine scheduling problem, and so would be NP-Hard (see [20], [42]). Since our time space is explicitly divided into periods, we must be more careful. Let us consider the following setting **PVSync-Sched** which corresponds to this restriction of **PVSync**:

PVSync-Sched

- **Inputs**:
  - The period set $\{1, \ldots, N\}$
  - The battery set **K** = $\{1, \ldots, K\}$
  - The job set **J** = $\{1, \ldots, J\}$: A job j requires t_j periods and we suppose that $\sum_j t_j \leq K.N$. No precedence constraints are imposed.
- **Ouputs**: We want to schedule the jobs of **J** inside the periods $1, \ldots, N$, in such a way that no more than K jobs are performed during a same period i.

Then we easily check that any instance of the well-known strongly NP-Complete **Bin Packing** problem (see [20], [42]) can be polynomialy reduced to an instance of **PVSync-Sched** so that **PVSync-Sched** is also strongly NP-Complete.

The *Battery* Level.
Let us consider now the *Battery* level, which means that we suppose that the jobs of **J** have been scheduled and that we deal with variables $Y_{j,k}$, that distribute the batteries among the jobs. We focus on the constraints (E3, E4, E5, E9, E10) of **PVSync**, without taking care of the activity of the PV-Plant, (infinite free production rates R_i), and while restricting ourselves to 2 batteries. More precisely, we consider the following restriction **PVSync-Battery** of **PVSync**:

PVSync-Battery

- **Inputs**:
  - The period set $\{1, \ldots, N\}$.
  - The job set **J** = $\{1, \ldots, J\}$: A job j requires e_j energy. Every job j has been scheduled inside a single period $i(j) \in \{1, \ldots, N-1\}$. We suppose that $J = 2.(N-1)$ and that no more than 2 jobs have been scheduled during a same period.
  - Two identical batteries k_1, k_2, initially loaded with a same energy amount H_0.
- **Outputs**: We want to assign the batteries to the jobs in a way that is consistent with the energy requirements.

Theorem 1: ***PVSync-Battery*** *is NP-Complete*.

Proof: We may consider the case when the initial load H_0 is equal to $\frac{\hat{e}}{2}$ with $\hat{e} = \sum_j e_j$. Then we see that for any period i, there must exist exactly 2 jobs $j(i)$ and $\bar{j}(i)$ which are scheduled at period i. We may suppose $e_{\bar{j}(i)} \geq e_{j(i)}$. It comes that k_1 and k_2 must be active during periods $1, \ldots, N-1$ and can only recharge at period N. During those periods $1, \ldots, N-1$, k_1 and k_2 must globally provide the same energy $\frac{\hat{e}}{2}$. Thus, solving our problem means partitioning the period set $1, \ldots, N-1$ into 2 subsets N_1 and N_2 in such a way that $\sum_{i \in N_1}(e_{\bar{j}(i)} - e_{j(i)}) = \sum_{i \in N_2}(e_{\bar{j}(i)} - e_{j(i)})$. We get a reduction to the **2-Partition** problem (see [20], [42]) and we conclude. **End-Proof**.

3.2 Merging the batteries: a projection mechanism

PVSync is a complex problem, with 2 encapsulated decision levels, respectively related to job scheduling and battery assignment, which both involve their own NP-Complete satisfiability sub-problems. The discussion about the variants of **PVSync** and Remark 4 showed that the production level may itself become more complex, even when remaining inside the **P-Time** class. So, even if we work according to the centralized paradigm, dealing with large size or real time constrained instances of

PVSync should push us to find a way to partially bypass some levels and among them the battery level. This trend will of course be reinforced as soon as we address the collaborative issue.

According to this purpose, we shortcut the battery level by merging the K batteries into a unique virtual one. Relaxing this way **PVSync** will later help us in a significant way, when it will come to the design of *job scheduler* oriented algorithms that schedule the jobs while partially bypassing the production issue. More precisely, we start from a **PVSync** instance and perform the following construction:

- We replace the K batteries, by a single *macro-battery* with storage capacity $\hat{C} = K.C$. For any period i, $\hat{Q}_i$ will mean the energy amount loaded into this *macro-battery* during period i.
- We forbid more than K jobs to be running during a same period and we impose the *Idle Battery* constraint (E13):
  For any period i,
  $\hat{Q}_i \leq C^R.(K–n(Z,i))$. (E13)
  This constraint bounds the energy loaded into the jobs without explicitly involving the batteries. It implies that no more than K jobs are simultaneously running.

Remark 5: The *macro-battery* is completely virtual and can be viewed as a *projection* of batteries onto the job scheduler. It provides us with a way to shortcut the battery level.

This construction leads us to set a relaxation of **PVSync**, denoted by **PVSync-Merge**. In order to analyze the complexity of the *Idle Battery* constraint (E13), let us restrict **PVSync** to 4 periods, unit-period jobs, no temporal constraints, and a PV-Plant activity reduced to energy purchase. More precisely, let us set the following problem **PVSync-Idle**:

PVSync-Idle

- **Inputs**:
  - The period set $\{1,2,3,4\}$.
  - The unit-period job set **J** = $\{1,\ldots,J\}$: A job j requires e_j energy.
  - *Battery* parameters K, C, C^R.
- **Ouputs**: We deal with a *macro-battery* initially loaded with an energy amount $H_0 = \frac{(\sum_j e_j)}{2}$, and provided with a storage capacity $\hat{C} = H_0$ and a *recharge* capacity coefficient $C^R = \frac{H_0}{J}$. This *macro-battery* can buy energy according infinite purchase costs in periods 1 and 3 and null purchase costs in periods 2, 4. Then we want to schedule under a null cost the jobs of **J** inside the periods $\{1,2,3,4\}$, in such a way that:
  - The *Idle Battery* constraint is satisfied: For any period i,
    $\hat{Q}_i \leq C^R.(K–n(Z,i))$. (E13)
  - For any period i, the load $\hat{W}_i$ of the *macro-battery* at the end of i does not exceed $\hat{C} = H_0$.

Then we check that solving **PVSync-Idle** means partitioning the job set **J** into 2 subsets J_1, J_2 such that $\sum_{j\in J_1} e_j = \sum_{j\in J_1} e_j$. We derive from the NP-Completeness of the 2-Partition (see [20], [42]) problem the NP-Completeness of the **PVSync-Merge** problem.
Of course, both the strong NP-Completeness of **PVSync-Sched** and the NP-Completeness of **PVSync-Idle** imply that **PVSync-Sched** is strongly NP-Hard.

Extending a Schedule Vector Z into a Feasible Solution of PVSync-Merge
Since the leader object in **PVSync-Merge** is the schedule vector $Z = (Z_{j,i}, j = 1,\ldots,J, i = 1,\ldots,N)$, we are now going to characterize the conditions which make possible to extend Z into a full feasible solution of **PVSync-Merge**, in a way which does not involve the variables $U, V, \hat{Q}, W$. This characterization will help us in dealing with *schedule* vector Z while bypassing the lower levels of **PVSync**. In order to provide it, we need some additional notations:

- For any periods i, i_1, we set:
  - $Conso(i, i_1, Z) = \sum_{j \text{ s.t. } i\leq Start(Z,j)\leq i_1} e_j$.
    $Conso(i, i_1, Z)$ means the energy consumption by the jobs which start no sooner than i and no later than i_1.
  - $ProdMax(i, i_1, Z) = \sum_{i_2 \text{ s.t. } i\leq i_2\leq i_1}(K - n(Z, i_1)).C^R$.
    $ProdMax(i, i_1, Z)$ means the maximal energy that the macro-battery may load during periods $i,\ldots,i_1$.

Then we may state:

Theorem 2. *Schedule vector Z may be extended into a feasible solution of the **PVSync-Merge** problem if and only if:*

- *For any period i:* $H_0 + ProdMax(1, i, Z) \geq Conso(1, i+1, Z)$. *(E14)*
- *For any periods i, i_1, s.t. $i \leq i_1$:* *(E15)*
  $\hat{C} + ProdMax(i, i_1, Z) \geq Conso(i, i_1 + 1, Z)$.

Proof: Above conditions (E14, E15) are clearly necessary: The first one tells us that at any period i, the macro-battery should have received enough energy in order to ensure that any job j starting no later than period $i+1$ might be achieved. The second one tells us that, if we consider a sequence of periods $i,\ldots,i_1$, then the macro-battery should receive enough energy during those periods in order to achieve all jobs starting no sooner than i and no later than

$i_1 + 1$. The key point is about sufficiency. We proceed by induction on i, and suppose the converse. We suppose that i_1 is such that we could schedule the production in such a way that all jobs starting no later than i_1 could be achieved, and that at the end of period i_1, we cannot provide the jobs j starting in $i_1 + 1$ with the energy that they require. We may impose our production strategy to be such that at any period i, we load the macro-battery with as much energy as possible, taking into account the *Idle Battery* constraint (E13) and the storage capacity $\hat{C}$. Then we see, by moving backward from i_1 to 1, that either we reach some period i such that (E15) is violated or we reach period $i = 1$ in such a way that (E14) is violated. We conclude. **End-Proof**

4 The cooperative issue

There are several way to address this *collaborative* issue. One may adopt the point of view of a specific player, and emulate the interaction that this player is likely to develop with its partners. Another approach consists in keeping on with the centralized paradigm, and supposing that some mediator player fairly distributes costs and profits among the other players. This second point of view corresponds to the *Cooperative Game* theoretical framework. Though our purpose here is to focus on the first approach, we are first going to briefly describe the way **PVSync** may be cast into the *Cooperative Game* framework.

4.1 A cooperative *PVSync* game

Let us recall that a *cooperative game* (Ω, Val) (see [33], [21]) is defined as a set of players Ω and a function Val which, to any subset A of Ω, called *coalition* makes correspond its *cost* value $Val(A)$. Then the problem becomes to fairly distribute the cost $Val(\Omega)$ among the players, in such a way that no coalition is tempted to leave the game. Several approaches may be tried. The most popular one is related to the *core* notion: a price vector $\pi = (\pi_\omega, \omega \in \Omega)$ is in the core of the game (Ω, Val) iff:

- $\sum_\omega \pi_\omega = Val(\Omega)$
- For any coalition A, $\sum_\omega \pi_\omega \leq Val(A)$

This core may be empty, which imposes trying weaker ways to distribute the cost $Val(\Omega)$ among the players, for instance according to the Shapley values. Conversely, it may contain too many elements, which leads to more restrictive notions such as the *Nucleolus* (see [7],[33]).

In the present case we suppose, for the sake of simplicity, that every job j is identified with exactly one player, and that the players are independent. This implies that no precedence relation $<<$ is imposed to the jobs, since such a constraint would induce a dependence between related players. Then we define a cooperative game *G-PVSync* by setting:

- $Val(A) = PVSync(\mathbf{J})$–$PVSync(\mathbf{J} - A)$. $Val(A)$ represents the marginal cost induced by the jobs of A with respect to $PVSync(\mathbf{J})$.

Then we may state:

Theorem 3: *If all sale prices S_i are null, then the core of G-PVSync is not empty*.

Proof: Let us recall that Bondareva/Shapley Theorem (see [8]) provides us with a characterization of the non-emptiness of the core of a cooperative game (Ω, Val). This characterization comes as follows:

- A non negative vector $\mu = (\mu_A, A \subseteq \Omega)$ is said to be *balanced*, if, for any player ω, $\sum_{A \text{ s.t. } \omega \in A} \mu_A = 1$.
- Then the cooperative game (Ω, Val) has a non-empty core if and only if for any balanced vector μ, $Val(\Omega) \leq \sum_A \mu_A Val(A)$. (E16)

So let us consider some balanced vector μ together with some optimal solution $Sol = (X, Y, Z, T, U, V, W, Q)$ of **PVSync**. Let us set $\overline{\mu} = \sum_A \mu_A$. (E16) requires:
$PVSync(\{1, \ldots, J\}) \leq \sum_A \mu_A (PVSync(\{1, \ldots, J\})$
$-PVSync(\{1, \ldots, J\} - A))$.
This equality is equivalent to: (E17)
$\sum_A PVSync(\{1, \ldots, J\} - A) \leq$
$(\overline{\mu} - 1).PVSync(\{1, \ldots, J\})$.
So we only need to check that it is possible to decompose $(\overline{\mu} - 1).Sol$ into a non negative linear combination $\sum_A \mu_A.Sol(\{1, \ldots, J\} - A)$, where every vector $Sol(\{1, \ldots, J\} - A)$ is a feasible solution of the restriction of **PVSync** to the jobs of $(\{1, \ldots, J\} - A)$. If we can do it, then we conclude since every value $PVSync(\{1, \ldots, J\} - A)$ is going to be no larger than the cost of $Sol(\{1, \ldots, J\} - A)$. Getting such a decomposition can be done by tracking inside Sol, for every job j, the energy consumed by j and the energy produced in order to match this consumption. As a matter of fact, we might also represent the way energy circulates according to (U, V, W, Q) as a flow vector in a network derived from the periods and the batteries involved in **PVSync** (*time expanded* network) and decompose it along the jobs of **J** into a multi-commodity flow vector (see [2], [9]). This process makes appear, for every battery k, the useless energy $C^{Res}(k)$ remaining inside k throughout the whole time space. It yields a decomposition of Sol as $Sol = \sum_j SolJob_j$, where every partial solution $SolJob_j$ meets reduced capacities $C - C^{Res}(k), k = 1, \ldots, K$. Then $Sol(\{1, \ldots, J\} - A)$ comes as $\sum_{j \in \{1, \ldots, J\} - A} SolJobt_j$. **End-Proof**

Remark 6: If we allow non-null sale price, then we may check that above reasoning does not hold anymore.

4.2 The job scheduler/PV-Plant interaction: surrogate estimators

We now address the collaborative issue while adopting the point of view of the job scheduler and trying to emulate the interaction that this player is likely to develop with its partner. This point of view corresponds to the most natural bi-level setting of **PVSync**. According to it, we rely on a surrogate estimator $SurrPVCost$, which with any schedule vector Z consistent with (E5, E12), and any value of some flexible parameter vector γ, associates an estimation $SurrPVCost(Z, \gamma)$ of the value $RestrictPVSync(Z)$. The role of the parameter γ is to introduce flexibility in order to mitigate the fact that $SurrPVCost(Z, \gamma)$ provides us with no more than an approximation of $RestrictPVSync(Z)$. The effective value of γ will be tuned all along the interaction process, in such a way that this process converges to a *consensual* solution. So, at every step during this process, we shall compute Z while restricting ourselves to contraints (E5, E12) together with some additional (surrogate) constraints $Cons(\gamma, Z)$. Those surrogate constraints will aim at both avoiding **Restrict-PVSync**(Z) to be unfeasible and at providing a container for the interaction between the job scheduler and its partner. We shall perform this computation of Z in such a way that it minimizes the cost $\alpha.(\sum_{j,i} i.Z_{j,i}) + SurrPVCost(Z, \gamma)$. Taken as a whole, this interaction will proceed as follows (see Fig. 4):

Collaborative algorithmic scheme

Initialize γ and constraints $Cons(\gamma, Z)$ about schedule vector Z and γ;

Set the initial proposal of the job scheduler: Compute Z which minimizes $\alpha.(\sum_{j,i} i.Z_{j,i}) + SurrPVCost(Z, \gamma)$ while meeting (E5, E12) and $Cons(\gamma, Z)$;

While Not $Stop$ do

1. Set the counter-proposal from the PV-Plant: Solve **Restrict-PVSync**(Z) and retrieve new constraints to insert into $Cons(\gamma, Z)$;
2. Update $Stop$; If Not $Stop$ then

   Update γ;

   Set the new proposal of the job scheduler: Compute Z minimizing $\alpha.\sum_{j,i} i.Z_{j,i} + SurrPVCost(Z, \gamma)$ while meeting (E5, E12) and updated $Cons(\gamma, Z)$;

Retrieve the best schedule vector Z obtained this way.

Of course, this heuristic management of **PVSync** may also be applied under the standard *centralized* paradigm approach in the case of large size instance or real time requirements.

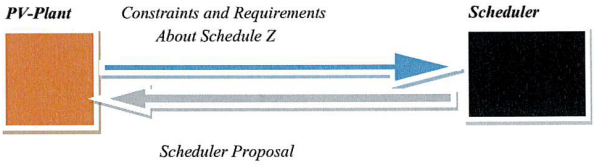


Figure 4: A collaborative scheme

4.3 Surrogate estimators and constraints

We distinguish the surrogate estimator $SurrPVCost(Z, \gamma)$ from the additional constraints $Cons(\gamma, Z)$.

Remark 7: Notice, in the case of the surrogate estimator $SurrPVCost(Z, \gamma)$, that it does not really aim at providing an approximation in the standard sense of $RestrictPVSync(Z)$, rather at efficiently driving above collaborative algorithmic scheme towards good solutions.

4.3.1 Surrogate constraints

Previous Section 3 leads us to insert (E14) and (E15) into $Cons(\gamma, Z)$. Still, though these constraints ensure the feasibility of the **PVSync-Merge** problem of Section 3, they may not be sufficient in the case of the full **PVSync** problem. Typically, if 2 jobs j_1 and j_2 such that $e_{j_1} = e_{j_2} = 10$, are scheduled to start at period 1, if $K = 2$ and if $H_{k_1}^{Init} = 5, H_{k_2}^{Init} = 15$, then we cannot successfully assign the batteries to the jobs, while we may extend this schedule into a feasible solution of **PVSync-Merge**.

In order to reinforce (E14, E15), we proceed in a heuristic way, while relying on the flexible vector γ:

- We first set, for any energy amount E : $m(E) = \lceil \frac{E}{C^R} \rceil$. This number $m(E)$ means the number of consecutive periods necessary in order to load a battery with E energy. Let also recall that $L(Z, i_1)$ denotes the number of idle batteries induced by Z at period i.
- Then we impose the following parametric heuristic constraints, which depend on flexible parameters γ_0 and γ_1:
  - For any period i, let $j_1, \ldots, j_S$ be the jobs scheduled to start at period $i+1$ according to schedule vector Z. Then, for any $i_1 \leq i$, we impose: (E18)
    $L(Z, i_1) \geq$
    $\gamma_0.|\{s \text{ s.t. } m(e_s \geq (i–i_1 + 1)\}|$.
  - For any period i, let us denote by $j_1, \ldots, j_S$ the jobs starting at $i + 1$, ordered according to decreasing $m(e_{j_s})$ values, and by $S_1 \leq S$ the largest s value such that $m(e_{j_{S_1}}) \geq i$. Then we impose that there exist $k_1, \ldots, k_{S_1}$ in **K** such that for any $s \leq S_1$:
    $HInit_{k_s} + i.C^R \geq \gamma_1.e_{j_s}$. (E19)

Constraints (E18, E19) aims at making in such a way that at any period i, there should exist batteries which have been idle for enough time in order to feed jobs starting in $i+1$.

4.3.2 A price based surrogate estimator

We proceed here in an empirical way, while following the idea that the production cost induced by a schedule vector Z is determined by the distribution of values $L(Z,i), i \in \{1,\ldots,N\}$. This suggests us to express the surrogate cost $SurrPVCost(Z,\gamma)$ as a sum $\sum_i \Pi_{i,L(Z,i)}$, where $\Pi_{i,L}$ is an estimation of the cost induced by L batteries in recharge (idle) at period i. According to this idea, we first make appear a *standard production price* $\Pi^{Stand}_{i,L}$ which, with any idle battery number L and any period i, associates a kind of a reference price of the recharge of L batteries at period i. We do it by noticing that if all batteries receive a same charge E^{Mean} at every period when they are idle, then the production cost $Cost$ should be equal to $\sum_i \Pi^{Stand}_{i,L(Z,i)}$. So we derive *standard production prices* $\Pi^{Stand}_{i,L}$ as follows:

$\Pi^{Stand}_{i,L} = P_i.(L.E^{Mean}–R_i)$ if $L.E^{Mean} \geq R_i$ and $\Pi^{Stand}_{i,L} = S_i.(L.E^{Mean}–R_i)$ else.

In order to flexibilize those prices, we follow intuition which tells us:

- If $L(Z,i).E^{Mean} \geq R_i$, then $\Pi_{i,L}$ should increase with P_i;
- If $L(Z,i).E^{Mean} \leq R_i$, then $\Pi_{i,L}$ should decrease (negative cost) as S_i increases.

This leads us to introduce 2 components γ_2 and γ_3 of flexible parameter γ and to set:

- $\Pi_{i,L} = \Pi^{Stand}_{i,L}.(1 + \gamma_2.(P_i–P^{Mean}))$ if $L(Z,i).E^{Mean} \geq R_i$,
- $\Pi_{i,L} = \Pi^{Stand}_{i,L}.(1 + \gamma_3.(S_i–S^{Mean}))$ else.

4.3.3 A machine learning based surrogate estimator

Instead of relying on energy price coefficients $\Pi_{i,L}$ we use a neural network CNN_PVSync in order to provide us with the quality of a scheduled vector Z. Network CNN_PVSync involves 467 synaptic coefficients and is trained with 9000 **PVSync**(Z) instances, among them 8110 training instances and 890 validation instances, solved with the MILP model of Section 2. This small ratio of 1/20 between the number of synaptic coefficients and the number of training instances eases the training process (stochastic gradient optimization process), making the error gap evolve in a monotonic way along the epochs and stabilize itself in a natural way in the neighborhood of some optimal error gap. The stochastic gradient algorithm behaves as if it were dealing with a standard optimization problem, with a small number of variables and an objective function defined by an average violation of a larger set of constraints. A consequence is that we do not need to observe the evolution of the error gap along the epochs in order to identify the epoch that induces the best error gap.

CNN_PVSync is designed as a *convolutional* neural network (CNN). CNNs have been mostly used for 2D-pattern recognition, since images are very large size inputs and since the convolutional masks are well-fitted to the recognition of local patterns. In the present case, our goal here is to learn the optimal value of a combinatorial optimization problem (the optimal cost related to the production sub-problem induced by fixing the decision of the job scheduler), in order to drive a heuristic scheduling process. An important feature of a CNN is that, at the contrary of most neural networks, it can deal with flexible inputs of different sizes. It is our case here, since the size of our target combinatorial optimization problem may vary. That is why we choose to work with a CNN. Notice that the error gap induced by the the CNN is not at stake here, rather its ability to drive the heuristic algorithm toward good solutions.

A convolutional network usually works in 3 steps. In the first step a same standard *perceptron* CM, called *convolutional mask*, is applied to fixed size segments of the input vector $IN = (IN_m, m \in \{1,\ldots,M\})$, where M is the variable size of the input data vector. This *perceptron* CM yields an output vector $OUT = (OUT_m, m \in \{1,\ldots,M\})$, with the same size as IN. In the next step, a *pooling* mechanism is applied to OUT, in order to compact it into the fixed size input vector $\overline{IN}$ of another perceptron $NPool$. In the last step, $NPool$ turns $\overline{IN}$ into the final output $\overline{OUT}$ of CNN. In the present case the final output of CN_PVSync is a number θ between 0 and 1. This number refers to the formulation of the optimal value $RestrictPVSync(Z)$ as a barycentric combination $LowPVSync(Z) + \theta.(UpPVSync(Z) - LowPVSync(Z))$, where $UpPVSync(Z)$ and $LowPVSync(Z)$ are respectively an upper bound and a lower bound of $RestrictPVSync(Z)$. In case $RestrictPVSync(Z)$ does not exist, that means in case **PVSync**(Z) is not feasible, we do as if $RestrictPVSync(Z)$ were equal to $UpPVSync(Z)$ (then θ should be equal to 1). Figure 5 shows the architecture of CNN_PVSync whose main components come as follows:

- **Input layer**: We homogenize any input $(Z,P,S,R,H^{Init},C,C^R,K)$ of **RestrictPVSync**(Z) as a $7(N+1)$ vector IN, with $IN[i] = (\overline{P}_i, \overline{S}_i, \overline{R}_i, \overline{\mu}_i, \overline{L}_i, \overline{C}, \overline{C^R})$, given by:
  - $\overline{P}_i = \frac{P_i}{P^{Mean}}; \overline{S}_i = \frac{S_i}{P^{Mean}}$.
  - $\mu_i = \sum_{j \text{ active at period } i} = e^{Mean}_j$; $\mu_0 = 0$; $\overline{\mu}_i = \frac{\mu_i}{R^{Mean}}$. This last quantity $\overline{\mu}_i$ identifies the normalized amount of energy consumed at any period.
  - $\overline{R}_i = \frac{R_i}{R^{Mean}}; \overline{R_0} = \frac{\sum_k H^{Init}_k}{R^{Mean}}$.

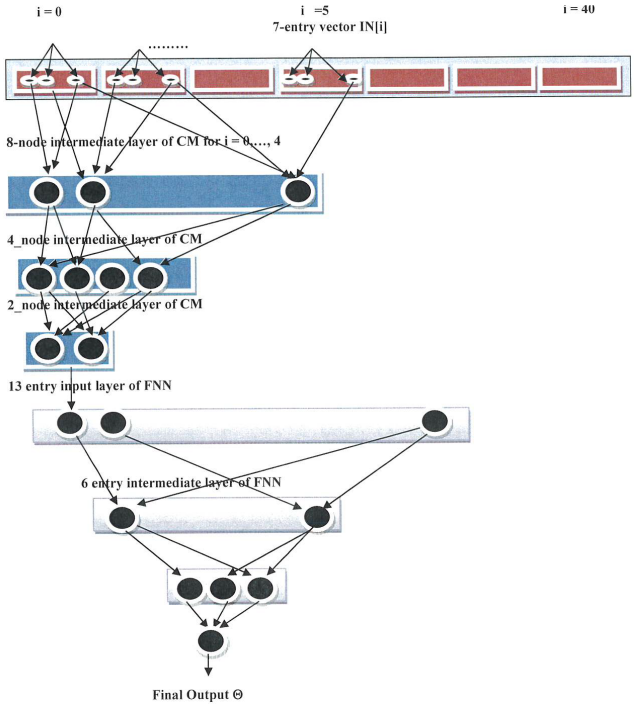


Figure 5: The neural network CNN_PVSync

- $\overline{L}_i = \frac{L(Z,i)}{K}$; $\overline{C} = \frac{C}{R^{Mean}}$; $\overline{C^R} = \frac{C^R}{R^{Mean}}$.

– **Convolutional Mask**: CM works on any vector $IN_i^* = (IN[i], \ldots, IN[i+4])$, which means an input with 35 input arcs. It contains 3 inner layers, respectively of sizes 8, 4 and 2, and ends into an output layer, that yields 1 input value OUT_i. This network is complete in the sense that all 322 inner synaptic arcs are allowed, together with standard biased sigmoid activation functions $x \to \frac{1}{1+exp(-kx)}$, with parameter k. Thus the number of synaptic coefficients related to this convolutional mask is 320 + 35 = 355.

– **The pooling Mechanism**: It works by merging consecutive values OUT_i into a single one, in such a way that we get an intermediate vector $\overline{IN}$, with 13 entries, all with values between 0 and 1.

– **The Final Perceptron** $NPool$: Once the pooling mechanism has been applied, we handle resulting 13 dimensional vector $\overline{IN}$ with a perceptron $NPool$, with input layer of size 13, intermediate layers of size 6 and 3, and a final layer of size 1. This network is complete in the sense that all 99 inner synaptic arcs are allowed, together with standard biased sigmoid activation functions $x \to \frac{1}{1+exp(-kx)}$, parameter k being provided with the same value as in the convolutional mask. Thus, the number of synaptic coefficients associated with this final perceptron is 13 + 99 = 112.

Taken as a whole, CNN_PVSync involves 467 synaptic coefficients.

– **Outputs of** CNN_PVSync: As previously told, the concatenation of CM and $NPool$ yields a value $\overline{OUT} = \theta$ between 0 and 1, which refers to the barycentric setting:
$RestrictPVSync(Z) = LowPVSync(Z) + \theta.(UpPVSync(Z) - LowPVSync(Z))$,
where $UpPVSync(Z)$ and $LowPVSync(Z)$ are respectively an upper bound and a lower bound of $RestrictPVSync(Z)$, computed as follows:

- $LowPVSync(Z) = -\sum_i S_i.R_i.$
- $UpPVSync(Z) = (\text{Sup}_i P_i).(\sum_j e_j) - (\sum_i S_i.R_i).$

In case **Restrict-PVSync**(Z) is unfeasible, we do as if $RestrictPVSync(Z)$ were equal to $UpPVSync(Z)$, that means is if θ were supposed to be equal to 1.

The way we train the network CNN_PVSync in order to make it learn its synaptic coefficients will be described in the next section 5, devoted to numerical experiments.

4.4 *Job scheduler* oriented heuristics for the search for schedule vector Z

Dealing with a blackbox objective function such that CNN_PVSync can only be done through a heuristic scheme. So we design a simple *job scheduler* oriented heuristic *SurrPVCost* that works in 2 steps and that we may adapt to both the price based version and the machine learning version of the $SurrPVCost$ estimator.

SurrPVCost Job Scheduler Oriented Algorithm

Initialization : Pick up the jobs j according to increasing Min_j values (updated through constraint propagation) and assign them starting periods which maintain constraints (E5, E12, E14, E15, E17, E18, E19), meet time windows and the precedence constraints, and minimize $\alpha.(\sum_{j,i} i.Z_{j,i}) + SurrPVCost(Z, \gamma)$;

While Not $Stop$ **do** : Remove some job j from the current schedule and reinsert it (while possibly delaying other jobs) in such a way that $\alpha.(\sum_{j,i} i.Z_{j,i}) + SurrPVCost(Z, \gamma)$ decreases.

5 Numerical experiments

Purpose: We want to evaluate the behavior of the *job scheduler* oriented heuristic *SurrPVCost*, implemented along the two surrogate estimators described in sections 4.3.2 and 4.3.3, and its ability to yield under small computational costs a good approximation of the optimal value of **PV-Sync**. According to this purpose, the MILP formulation of **PV-Sync** is used only in order to provide us with benchmark results.

Technical Context: We use a processor IntelCore i5-6700@3.20 GHz, with 16 Gb RAM, together with a C++ compiler, Linux as O.S. libraries CPLEX20.1 (for ILP models) and TensorFlow/Keras (for machine learning).

Instances: As for the PV-Plant side, we generate 2 integers M and N, N being a multiple of M, with $N = 10, \ldots, 40, M = 2, \ldots, 5$. Then we split the period set into M macro-periods, corresponding to different mean production rates and prices. Related coefficients R_i, P_i, S_i are generated accordingly. Introducing those macro-periods provides us with realistic prices and production levels that may be related to human activity and to the weather.
As for the scheduling part, we try to both ensure feasibility and put stress on the instances. So we generate J, together with mean duration and mean energy coefficients t^{Mean} and E^{Mean}. We set $C = 2.\text{Sup}_j\ e_j$ and $K = \lambda.J.(\frac{t^{Mean}}{N})$, where λ is some control parameter. For any $k = 1, \ldots, K$, we generate H_k^{Init} between $\frac{C}{3}$ and C. In order to make the PV production match the demand, we update the R_i by doing in such a way that $\sum_i R_i = \tau.J.C$, τ being a control parameter with value between 0.5 and 2. Finally we generate the key parameter C^R in such a way that batteries globally receive at least $\beta.J.C$ energy units during the whole process, β being a control parameter with value between 1.5 and 4. So the main parameters of an instance are: N = *Period Number*, J = *Job Number*, M = *Macro-period Number*, K = *Battery Numbe*r, t^{Mean} = *Mean Job Length*, α = *Time versus Money Value*, β = *Recharge Stress Value*, τ = *Production Stress*.

According to this, we generate 10 groups of instances, every instance group G_Id containing 30 instances consistent with the characteristics N, J, M, t^{Mean}, K, α, β, τ described in the following table 3.

Table 3: Instance group characteristics

G_Id	N	J	M	t^{Mean}	K	α	β	τ
1	40	21	3	4	4	1	2	0,5
2	40	23	4	5	4	0,5	3	1
3	40	20	5	6	5	0,2	4	2
4	40	24	3	4	5	1	2	0,5
5	40	32	4	6	4	0,5	3	1
6	40	34	5	8	4	0,2	4	2
7	60	43	4	5	3	1	3	1
8	60	47	6	10	3	0,5	4	2
9	60	53	4	5	5	1	3	1
10	60	61	6	10	5	0,5	4	2

5.1 Training the neural network CNN_PVSync.

For every group instance G_Id described above and for every instance Id in G_Id , we randomly generate 30 schedule vectors Z, which meet the constraints (E5, E12, E14, E15, E17, E18, E19). We do it by turning the initialization procedure of the *SurrPVCost* Algorithm into a partial enumeration procedure, in such a way it can randomly generate several vectors Z meeting constraints (E5, E12, E14, E15, E17, E18, E19). Notice that those vectors may not be extended into a feasible solution of **PVSync** and that in such a case we consider that $RestrictPVSync(Z)$ is equal to $UpPVSync(Z)$, requiring related value θ to be equal to 1. Then we split resulting set of 9000 instances between a training set of 8110 instances and a validation set of 890 instances. The high ratio of 20 between the number of instances and the number of synaptic coefficients significantly eases the training process, making the error gap evolve in a monotonic way until stabilizing itself in a natural way around some approximation of the optimal error gap. We perform the training process while testing 12 sets Hy of hyperparameters, described in the following Table 4. Those hyperparameters are the parameter k of the sigmoid activation function, the batch size Ba, the number of epochs Ep, the loss formula Los (among MS = Mean Square Error, MA = Mean Absolute Error, MSL = Mean Square Logarithmic Error), the optimizer Op (among Ad = *Adam*, Nad = *Nadam*, Amx = *Adamax*, RMS = *RMSprop*). This table also makes appear, for every hyperparameter set Hy, the gaps (in %) between $RestrictPVSync(Z)$ and $CNN\_PVSync(Z)$, computed by referring to the value θ. Gap_T means here the value (in %) of this gap related to the training instance, and Gap_V means the same value (in %) related to the validation instances.

Table 4: hyperparameter sets

Hy	k	Ba	Ep	Los	Op	Gap_T	Gap_V
1	1	2	30	MS	Ad	9.6	12.9
2	0.5	4	30	MS	Ad	11.2	14.8
3	2	1	30	MS	Ad	9.4	12.1
4	3	2	30	MS	Amx	10.5	14.0
5	1.5	4	30	MS	Nad	8.6	11.3
6	2.5	8	30	MS	RMS	9.1	12.5
7	1.5	4	50	MA	Ad	8.7	11.8
8	1.5	4	50	MA	Ad	8.1	11.0
9	1.5	4	50	MSL	Ad	8.4	11.2
10	1.5	2	50	MSL	Nad	8.7	11.9
11	1.5	4	50	MA	Amx	10.9	14.7
12	1.5	8	50	MS	RMS	8.3	11.1

Comments: The best gap between $RestrictPVSync(Z)$ and $CNN\_PVSync(Z)$ corresponds to $Hy = 8$, with value a little bit larger than 8% for the training instances and 11 % for the validation instances. Recall that this error gap is not really at stake, rather the ability of $CNN\_PVSync(Z)$ to drive the heuristic *job scheduler* oriented *SurrPVCost* algorithm towards satisfactory solutions.

5.2 Evaluating the heuristic management (*job scheduler* oriented algorithm *SurrPVCost* of Section 4.4) of the PVSync problem through surrogate components

For any instance group G_Id described in Table 3, we generate exactly one instance Id and, for every such instance:

- We apply the CPLEX12 library (Table 5) to the **PVSync_MILP** model. Then we get (in less than 1 CPU h), a lower bound LB, an upper bound UB and CPU time T_MILP.
- We apply (Table 6) the *job scheduler* oriented algorithm *SurrPVCost* of Section 4.4 while relying on the pricing mechanism described in Section 4.3.2 and setting $\Pi_{i,L} = \Pi_{i,L}^{Stand}$ for any i, L. We denote by $PR1$ resulting **PVSync** value. We do the same (Table 6) with 8 combinations of γ values and denote by $PR8$ resulting value. We provide the CPU times T_PR8 (in seconds) required by the scheduling heuristic *SurrPVCost* described in Section 4.4.
- We apply (Table 7) the *job scheduler* oriented algorithm *SurrPVCost* of Section 4.4 while relying on the machine learning estimator described in Section 4.3.3 and denote by ML related value. We provide the CPU times T_ML required by the *SurrPVCost job scheduler* oriented algorithm described in Section 4.4.

Those results may be summarized into the following tables:

Table 5: Behavior of the **PVSync_MILP** model

Id	LB	UB	T_MILP
1	- 239.64	- 235.91	3600
2	- 85.56	- 82.47	3600
3	- 1439.38	- 1311.60	3600
4	488.45	619.15	3600
5	212.90	248.33	3600
6	177.46	198.35	3600
7	1738.56	1760.33	3600
8	$Fail$	$Fail$	$Fail$
9	571.31	3742.10	3600
10	- 52.92	155.27	3600

Comments: As expected, the global ILP model is in trouble, even on small instances. We notice that we may get negative cost values, because of the sales. This makes difficult reasoning in terms of percentages. Still, for small J, UB looks close to optimality.
In case the *job scheduler* oriented heuristic *SurrPVCost* relies on the parametric pricing mechanism described in Section 4.3.2, it seems to be rather efficient. Though it only relies on a heuristic *SurrPVCost* algorithm for the computation of schedule vector Z, it yields better results than the MILP model in the 3 instances with largest sizes. Its

Table 6: Behavior of the pricing scheme

Id	UB	$PR1$	$PR8$	T_PR8
1	- 235.91	- 216.68	- 216.68	48.22
2	- 82.47	- 75.84	- 78.52	71.83
3	- 1311.60	- 1165.59	- 1208.45	76.82
4	619.15	659.07	659.07	62.64
5	248.33	406.56	327.05	131.83
6	198.35	221.16	200.04	83.49
7	1760.33	1971.18	1904.09	579.50
8	$Fail$	- 397.6	- 397.6	1000,6
9	3742.10	1267.59	1267.59	1008.17
10	155.27	167.52	139.68	478.62

Table 7: Behavior of machine learning

Id	UB	ML	T_ML (s)
1	- 235.91	- 193.38	10.85
2	- 82.47	- 50.08	21.28
3	- 1311.60	- 890.00	28.60
4	619.15	766.79	15.50
5	248.33	398.20	45.60
6	198.35	254.06	25.08
7	1760.33	1861.24	72.25
8	$Fail$	- 359.07	105.10
9	3742.10	1548.25	180.05
10	155.27	199.00	80.05

sensitivity to parameter γ does not look very significant: In 4 cases, the best result is obtained with standard prices $\Pi_{i,L} = \Pi_{i,L}^{Stand}$ for any i, L. In case we apply *SurrPVCost* algorithm while relying on the machine learning oriented estimator described in Section 4.3.3, we see that the gap between UB and ML is more important. Yet, this second approach offers more genericity features than the first one.

6 Conclusion

We dealt here with a complex scheduling problem, involving encapsulated resources and synchronization mechanisms. Since handling it as a whole is difficult and may not fit collaborative contexts, we shortcut the resource production sub-problem and replaced it by a parametric surrogate estimator. We tried 2 approaches: the first one relied on artificial prices; the second one on a neural network. Numerical experiments made appear, at least in our case, a better efficiency of the first approach. Still, because of the genericity of machine learning, a priori better fitted for the management of the uncertainty inherent to our problem, it would be worthwhile to try to go deeper with it. Those two points will be the backbone of a future research.

Acknowledgement: Present work was funded by French ANR: *National Agency for Research*, by Labex IMOBS3 and by PGMO Program.

References

[1] Adulyasak Y., Cordeau J. F., Jans R.: The production routing problem: A review of formulations and solutions. *Computers and Operations Research*, 55, p 141-152, (2015). https://doi.org/10.1016/j.cor.2014.01.011.

[2] Ahuja R.K., Magnanti T.L., Orlin J.B., Reddy M.R.: *Applications of network optimization*. Handbook of Operation Research and Management Sci. 7, p 1-83, 1995.

[3] Albrecht A., Pudney P.: *Pickup and delivery with a solar-recharged vehicle*. Ph.D. thesis Australian Society for O.R (2013). www.asor.org.au/conferences/asor2013.

[4] Almuhtady A., Lee S., Romeijn E., Wynblatt M., Ni J.: A degradation informed battery swapping policy for fleets of electric or hybrid electric vehicles. *Transportation Science* 48, 4, p 609-618 (2014). https://doi.org/10.1287/trsc.2013.0494.

[5] Balbiyad S.: *Collective self consumption: computing the optimal energy distribution coefficients considering local energy management*. ENSTA/EDF OSIRIS, (2019).

[6] Bendali F., Mole Kamga E., Mailfert J., Quilliot A., Toussaint H.: *Synchronizing Energy Production and Vehicle Routing*. RAIRO-O.R, 55 (4),pp. 2141-2163. (2021). https://doi.org/10.1051/ro/2021093.

[7] Berge C. : *Théorie des Jeux à n personnes* . Gauthier-Villars, Paris, Memorial Sciences Maths 138, (1957). http://www.numdam.org/item/?id=MSM_1957_138_1_0.

[8] Bondareva O. N. : Some applications of linear programming methods to the theory of cooperative games. *Problemy Kibernetica* 10, p 119-139, (1963).

[9] Bsaybes S., Quilliot A., Wagler A.: Vehicle fleet management using flows in time-expanded networks. *TOP*, p 1-24, (2019). DOI: 10.1007/s11750-019-00506-4

[10] Caprara A., Carvalho M., Lodi A., Woeinger G. J.: A study on the complexity of the bilevel knapsack problem. *SIAM Journal on Optimization* 24 (2), p 823-838, (2014).

[11] Chen L., Zhang G.: Approximation algorithms for a bi-level Knapsack problem. *Theoretical Computer Sciences* 497, p 1-12, (2013). https://doi.org/10.1016/j.tcs.2012.08.008

[12] Chen L., Englund C.: Cooperative intersection management: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 17-2, p 570-586, (2016). https://doi.org/10.1109/TITS.2015.2471812

[13] Chen Z. L.: Integrated production and distribution scheduling: Review and extensions. *Operations Research* 58, p 130-148, (2010). https://doi.org/10.1287/opre.1080.0688.

[14] Chen Z. L., Vairaktarakis: Integrated production and distribution operations. *Management Science* 51, p 614-628, (2005). DOI: 10.1287/mnsc.1040.0325

[15] Chretienne P., Hazir O., Khadad-Sidhoum S.: Integrated batch sizing and scheduling on a single machine. *Journal of Scheduling* 14-6, p 541-55, (2011). DOI : 10.1007/s10951-011-0229-x

[16] Colson B., Marcotte P., Savard G.: Bi-level programming: A survey. *4OR* Vol 3 (2), p 87-107, (2005). DOI:10.1007/s10288-005-0071-0.

[17] Deb S., Tammi K., Kalita K., Mahanta P.: Impact of electric vehicle charging station load on distribution network; *Energies* 11 (1), p 178-185, (2018). https://doi.org/10.3390/en11010178

[18] Dempe S., Kalashnikov V., Perez-Valdez G., Kalashikova N.: *Bi-level Programming Problems Theory*, Springer (2015). ISBN 3662458276, 9783662458273

[19] Erdelic T., Caric T.: A survey on the electric vehicle routing problem. *Journal of Advanced Transportation*, (2019). https://doi.org/10.1155/2019/5075671.

[20] Garey M. R., Johnson D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Co. Ed. (1979). ISBN 0-7167-1044-7

[21] Granot D., Granot F.: On some network flow games. *Maths O.R* 17, p 792-841, (1992). https://doi.org/10.1287/moor.17.4.792

[22] Grimes C., Varghese O., Ranjan S.: *Light, water, hydrogen: The solar generation of hydrogen by water photoelectrolysis*. Springer-Verlag US, (2008). • DOI:10.5860/choice.45-6194

[23] Hall N. G., Potts C. N.: The coordination of scheduling and batch deliveries. *Annals of Operations Research*, 135, p 41-64, (2005). DOI https://doi.org/10.1007/s10479-005-6234-8.

[24] Hartman S., Briskorn D. : An updated survey of variants and extensions of the resource-constrained project scheduling problem. *EJOR* 297, 1, p 1-14, (2022). https://doi.org/10.1016/j.ejor.2021.05.004.

[25] Irani S., Pruhs K.: Algorithmic problems in power management. *SIGACT News*, 36, 2, p 63-76, (2003). DOI:10.1145/1067309.1067324.

[26] Kleinert T., Labbé M., Ljubic I., Schmidt M.: A survey on mixed integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization* 9, 21 p, (2021). https://doi.org/10.1016/j.ejco.2021.100007.

[27] Koc C., Jabali O., Mendoza J., Laporte G.: The electric vehicle routing problem with shared charging stations. *ITOR*, 26 , p 1211-1243, (2019). doi:https://doi.org/10.1111/itor.12620.

[28] M.Krzyszton M.: Adapative supervison: method of reinforcement learning fault elimination by application of supervised learning. *Proceedings of the 2018 FEDCSIS AI Conference*, p 139-149, (2018). DOI: http://dx.doi.org/10.15439/978-83-949419-5-6.

[29] J. Kumar J., Ranga V.: Multi-robot coordination analysis, taxonomy and future scope. *Journal of Intelligent and Robotic Systems*, 102:10, (2021). DOI: 10.1007/s10846-021-01378-2

[30] Luthander R., Widen J., Nilsson D., Palm J.: Photovoltaic self-consumption in buildings: A review. *Applied Energy* 142, p 80-94, (2015). DOI: 10.1016/j.apenergy.2014.12.028

[31] Macrina G., Pugliese L.D, Guerriero F.: *The green-vehicle routing problem: A survey*. In Modeling and Optimization in Green Logistics, Cham: Springer International Publishing. p. 1-26, (2020). DOI: https://doi.org/10.1007/978-3-030-45308-4_1.

[32] Orji.M.J, Wei.S. Project Scheduling Under Resource Constraints: A Recent Survey. *Inter. Journal of Engineering Research and Technology* (IJERT) Vol. 2 Issue 2, (2013). DOI : 10.17577/IJERTV2IS2508.

[33] Owen G. : *Game Theory*.Academic Press, (1982). ISBN 0125311508, 9780125311502.

[34] Rizk Y., Awad M., Tunstel E.: Cooperative heterogenous mutlti-robot systems: a survey. *ACM Computing Surveys* 29, (2019). DOI : https://doi.org/10.1145/3303848.

[35] Sarmiento A.M., Nagi R.: A review of integrated production-distribution systems. *IEE Transactions* 31, 1061-1074, (1999). DOI : https://doi.org/10.1023/A:1007623508610.

[36] Smarter Together: Reports on collective self-consumption of photo-voltaic. *Technical Report Smarter Together*, (2016). `https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5adb475db\&appId=PPGMS`

[37] Souffran G., Liegeville L., Guerin P.: Simulation of real world vehicle missions using a stochastic Markov chain model for optimal power train sizing. *IEE Transactions on Vehicular Technology* 61, 8, p 3454-3465, (2012). https://doi.org/10.1109/TVT.2012.2206618

[38] Tamiselvi S., Gunasundari S., Karuppiah N., Razak A., Madhusudan S., Nagarajan V., Sathish T., Shamim M., Saleel A., Afzal A.: A review of battery modelling techniques. *Sustainability* 13, p 2-26, (2021). https://doi.org/10.3390/su131810042

[39] Tremblay O., Dessaint L.: Experimental validation of a battery dynamic model for E.V applications. *World Electric Vehicle Journal* 3, (2009). https://doi.org/10.3390/wevj3020289

[40] Trotta M., Archetti C., Feillet D., Quilliot A.: A pickup and delivery problem with electric vehicles and local energy production. *Proc. Triennial Symposium on Transportation Analysis* (TRISTAN), 6 pages, (2022). https://hal.science/hal-03723282v1

[41] Verma A.: Electric vehicle routing problem with time windows, recharging stations and battery swapping stations. *Euro Journal of Transportation Logistics* 7, p 415-451, (2018). https://doi.org/10.1007/s13676-018-0136-9.

[42] Wegener I.: *Complexity Theory*. Springer (2005). Wegener I.: *Complexity Theory*. Springer (2005).

[43] Wojtuziak J., Warden T., Herzog O.: Machine learning in agent based stochastic simulation: Evaluation in transportation logistics. *Computer and Mathematics with Applications* 64, p 3658-3665, (2012). https://doi.org/10.1016/j.camwa.2012.01.079.

[44] Zsiboracs H., Baranyai H., Vincze A., Haber I., Pinter G.: Economic and technical aspects of flexible storage photovoltaics systems in Europe. *Energies* 11, p 1445-1450, (2018). https://doi.org/10.3390/en11061445

Exploring the Power of Dual Deep Learning for Fake News Detection

Hounaida Moalla[1,2,*], Hana Abid[1], Dorsaf Sallami[3], Esma Aïmeur[3] and Bassem Ben Hamed[2]
[1]Department of Technological Information, ISET, Sfax, Tunisia
[2]Department of Mathematics and Business Intelligence, ENET'Com, Sfax, Tunisia
[3]Department of Computer Science and Operations Research, University of Montreal, Montreal, Quebec, Canada
hounaida.moalla@isetsf.rnu.tn, hanaabid@outlook.com, dorsaf.sallami@umontreal.ca,
aimeur@iro.umontreal.ca, bassem.benhamed@enetcom.usf.tn

Keywords: Fake news, survey, deep learning, transformers, generative artificial intelligence, detection, personality traits, big five, SEM

Received: April 2, 2024

The rise of social media has intensified the spread of fake news, a problem further exacerbated by generative artificial intelligence (AI). Hence, the need for improved detection of both human-created and AI-generated fake news using advanced AI models is critical. This paper proposes a survey to assess knowledge and attitudes towards news and AI, combining demographic data, personality traits, and the ability to distinguish between real and AI-generated news. Additionally, we create a new dataset, ERAF-News, containing real, fake, AI-generated true, and AI-generated fake news. To classify different types of news, we developed a dual-stream transformer model, DuSTraMo. This model leverages the capabilities of two parallel transformers to enhance the accuracy of news classification. The survey, involving 83 participants from 9 countries, revealed that respondents struggle to differentiate human-generated from AI-generated news. Notably, BERT outperformed GPT-2 and BART in generating realistic text, and RoBERTa and DistilBERT achieved over 98% accuracy in fake news classification. Dual-GPT models also showed high accuracy. This study underscores the effectiveness of the DuSTraMo model and the ERAF-News dataset in enhancing the detection of both human-created and AI-generated fake news. The findings highlight the increasing dominance of AI in this domain and the pressing need for advanced methods to combat fake news. Additionally, a survey examining users' responses to fake news reveals a concerning inability to accurately identify false information.

Povzetek: Članek raziskuje zaznavanje lažnih novic z uporabo dvojnoga globokega učenja. Predstavlja nov model DuSTraMo in zbirko podatkov ERAF-News za boljšo detekcijo človeških in AI-generiranih lažnih novic.

1 Introduction

The prevalence of fake news in the modern media environment has grown significantly. It becomes now more difficult than ever to differentiate correct information from false one owing to the fast growth of social media and online platforms [1]. The effects of propagating disinformation are serious, with the ability to change public perception, have an impact on political and social decisions [2], and cause confusion among individuals [3]. The free dissemination of false news can mislead the target audience or gain notoriety or financial advantage. [4, 5].

In the contemporary landscape, interaction with information has undergone a significant evolution, largely attributed to the proliferation of social media and online platforms. Historically, the predominant sources of information were traditional media outlets such as print newspapers [1], radio, and television, all of which adhered to stringent verification protocols.

However, the advent of platforms such as Facebook and Twitter, recently renamed X[2] [3], has revolutionized the way information is disseminated, enabling individuals worldwide to instantaneously share knowledge. Certainly, social media platforms play a significant role in the rapid dissemination of misinformation, complicating the assessment of online information's credibility [6]. This transformative shift not only reshapes the dynamics of information transmission but also underscores the paramount importance of media literacy in discerning truth amidst the continuous influx of online information [7].

Within the persistent issue of fake news, scholars are earnestly committing their efforts to the development of robust methods aimed at identifying and mitigating misleading information [8, 9, 10].

In the literature, several works have been carried out on Fake news detection. These works have varied both on the strategic and technical side and particularly affected

[1] https://rb.gy/5493gt, Last access: 18 Jully 2023

[2] Throughout the rest of the paper, the terms Twitter and X will be used interchangeably.

[3] https://rb.gy/98bc01, Last access: 02 November 2023

three main areas. Firstly, the use of surveys such as [11, 12, 13] have been based either on the analysis of the personality traits of respondents, or the mechanism of circulation of fake news, but have not discussed the degree of awareness of netizens to distinguish between types of news. Secondly, the automatic detection of fake news using ML methods [14, 15, 16] or transformers [15] used simple architectural models. The only recent paper that used dual-stream was [17] but simply duplicated the BERT model in both streams. Thirdly, processing datasets that have just two classes of news (real or fake) [18, 19] besides few works that added the third class of generated news [20, 21, 22]. The authors of [23] maked a multi-classification of information by treating degrees of falsehood such as True, MostlyTrue, Half-True, Barely-True, False, and Pants-on-Fire. No papers proposed to make a classification of four classes (real, fake, generated real and generated fake) and no dataset exists that contains all four classes.

In this paper, the fake news issue is addressed in-depth through the seamless integration of a survey and an artificial intelligence (AI) method. Firstly, a survey is used to delve deep into the public's attitudes. This survey tends to offer quantitative data that explores the nature and extent of the influence of fake news on the behavior of Internet users. Secondly, the capabilities of AI were leveraged, specifically deep learning based on transformers, for the detection of fake news. The innovation in this approach lies in its objective to differentiate between authentic news, false information, and content generated by AI (GAI).

To the best of our knowledge, this is the first paper that explores the synergy between these two approaches and allows for a more thorough examination of the intricate dynamics surrounding fake news, identification, and societal impact.

Hence, the key contributions of this paper are:

- **Diffusing a survey named "Generation of Fake News":** This questionnaire aims to investigate users' personalities, understanding of AI, awareness of fake news, and their capability to distinguish between fake and real news.
- **Creating a new dataset named "Extended Real And Fake News Dataset (*ERAF-News*):** The dataset includes real news, fake news, and real and fake news generated by AI. It can be used to test and refine state-of-the-art algorithms to detect fake news. The idea behind adding the fourth class is to distinguish between generated Real and generated Fake. This allows to optionally accept generated Real News and reject generated Fake ones.
- **Proposing a new Dual-Stream Transformers Model *(DuSTraMo)*:** This architecture aims to exploit complementary capabilities from two parallel models, potentially leading to improved performance on various natural language processing tasks.

The remaining parts of his paper proceed as follows: Section 2 provides an overview of three key areas: (a) prior survey-based research on fake news; (b) studies employing deep learning techniques based on transformers to classify the news; and (c) dynamic landscape of generative AI tools and their impact on fake news generation. Section 3 offers an overview of transformers. Section 4 intricately delves into the specific contributions and explains the methodology. It includes (a) an in-depth exploration of the survey, covering design rationale, section selections, presentation of collected data, and the chosen evaluation method; (b) a comprehensive account of the creation of the novel *ERAF-News* dataset; and (c) an exposition of the innovative *DuSTraMo* architecture designed for the detection of fake news. Section 5 offers an in-depth exploration of the obtained results, encompassing: (a) the findings from the investigation; (b) a performance evaluation of the *ERAF-News* dataset generation; and (c) a comprehensive discussion of the results derived from the deep learning models developed.

2 Related works

This section presents the literature in three distinct areas: (1) research focused on surveys [4] in the fake news context, (2) generation process based on a few pre-trained models, and (3) new method for fake news detection.

2.1 Surveys for fake news catch

The fast transmission and persistence of misinformation online pose a multifaceted threat [25, 24], impacting various domains including politics [27, 26], culture [29, 28], finance [31, 30], and psychology [24]. For the purpose of evaluating the accuracy of Internet users' answers, conducting an online survey can prove invaluable for assessing the attitudes, cultural orientations, and more of Internet users, thereby measuring the consistency of their responses.

The authors [11] are interested in the different mechanisms of circulation of disinformation on online social networks, and propose avenues for identification and in-depth analysis.

Linguistic analysis of information plays a principal criterion in the detection of false information. For this, one approach is to use a survey comprising a series of questions, [5], which can range from general queries to personalized or domain specific.

The objective of a survey [32] is to explore the correlation between the optimal user experience while navigating social media platforms, the behavior of sharing fake news, online trust, and heightened social media consumption. This research draws upon pertinent studies from the literature on fake news, online trust, and social media usage to inform the development of the questionnaire. Table

[4] Throughout the rest of the paper, the terms survey and questionnaire will be used interchangeably.

[5] https://shorturl.at/aIKP6, Last access: 02 September 2023

Table 1: Comparison of some fake news surveys

Ref.	Title	Subject	Nb responders
[33]	The role of social media in spreading panic among primary and secondary school students during the COVID-19 pandemic: An online questionnaire study from the Gaza Strip, Palestine	COVID-19	942
[34]	Surveying fake news: Assessing university faculty's fragmented definition of fake news and its impact on teaching critical thinking	Fake news	69
[35]	The Influence of Political Ideology on fake news Belief: The Portuguese Case	Fake news on political ideology	712
[36]	Fake news: the impact of the internet on population health.	Fake news of health information	1195
[37]	fake news Reaching Young People on Social Networks: DistrustChallenging Media Literacy	Fake news	408
[38]	Fake or real news? Understanding gratifications and personality traits of individuals sharing fake news on social media platforms	Fake news	221
[39]	The Role of Risk Perception and Ability to Detect fake news in Acceptance of COVID-19 Vaccine among Students of Shiraz University.	COVID-19	382
[40]	Anger makes Fake news viral online.	Fake news (offline)	1291
[41]	Examining the role of emotions, sharing motivations, and psychological distance of COVID-19-related fake news	COVID-19	150 toilet paper shortage-related
			149 celebrity scandal rumors

1 provides a brief overview of the studies on fake news surveys.

Comprehending the elements that influence how individuals react to fake news is vital for devising efficient strategies to minimize its detrimental impact on society. Personality traits play an essential role to shape people's behaviors[12, 13]. The Big Five model [6] seems a highly valuable standard for comprehending personality. This model is commonly named OCEAN (Openness, Conscientiousness, Extroversion, Agreeableness and Neuroticism), derived from its five key dimensions. It assesses an individual's position on each dimension, providing valuable insights into their personality traits and behavioral characteristics.

2.2 Methods for detecting fake news

Popular techniques for fake news detection, such as source verification, fact-checking, and cross-checking [42] are important to assess the credibility of news stories [43]. However, deeper analysis based on AI and machine learning (ML) algorithms has made it possible for sentiment analysis and language pattern identification, letting specialists to identify disinformation [44, 45].

Academic studies propose diverse techniques for automatic fake news detection, primarily relying on content-based techniques [46, 47], natural language processing (NLP) [18, 48], or data mining [49], ML methods [14, 15], or the use of social context-based techniques, as discussed in [50]. Additionally, some models adopt a hybrid integration of both approaches, as demonstrated by [51].

For assessment and classification of information, the use of NLP techniques and AI methodologies such as random forests (RF) [52], support vector machines (SVM) [53], long short-term memory (LSTM) neural networks [54], and transformers [19, 17, 55, 56], allows information professionals to enhance their ability to detect fake news effectively.

Researchers in [16] propose an innovative language-agnostic technique based on text attributes to discern fake news using various datasets. The outcomes reveal that the RF and SVM algorithms achieve accuracy of around 88% and 89% respectively, when applied to the FakeBr-

[6] https://shorturl.at/abFS4, Last access: 25 September 2023

Table 2: A comparison between fake news detection approaches

Date	Ref.	Dataset	Dataset classe			Models
			Real	Fake	GAI	
2019	[20]	RealNews(Common Crawl)	x	x	x	GPT2, BERT, Grover-Mega
2020	[18]	FakeOrRealNews	x	x		ML models
	[19]	FakeOrRealNews	x	x		BERT, RNN
	[16]	TwitterBR, FakeBrCorpus, FakeNewsData1, FakeOrRealNews, btvlifestyle	x	x		ML models
	[21]	newsQA dataset extension	x	x	x	Grover-Mega, Zero-shot
2021	[15]	Fake or real news, Combined corpus	x	x		ML models, CNN, LSTM, HAN, BERT, RoBERTa, DistilBERT, Electra
	[22]	S, xl, s-k, xl-k, Webtext, GPT3-WebtextrealNews	x	x	x	GPT2, GPT3, Grover
2022	[42]	Debate , PHEME	x	x		NLP framework
	[43]	Amazon, Common Crawl, Fake and Real News	x	x		Proposed pipline
	[44]	Created dataset	x	x		ML models
	[52]	FakeNewsNet	x	x		ML models
	[62]	RAWFC, LIAR-RAW	x	x		ML models, CNN, RNN, SentHAN, DeClarEdEFEND, SBERT-FC, GenFE, GenFE-MT, CofCED
2023	[17]	FakeNewsNet	x	x		Dual BERT
	[56]	Private dataset	x	x		Keyword-based, Rule-based, ML models
	[63]	CovidNews +NQ-1500	x	x	x	BM25, GPT-3.5, GENREAD, REIT, CTRLGEN, REVISE, RoBERTa-based

Corpus dataset [16]. Furthermore, [56] presents a machine-learning-centered method. Emphasizing the significance of source reliability, this technique attains a commendable accuracy rate of 90%. Table 2 provides a concise overview of the primary studies conducted in the literature concerning the automated detection of fake news [7].

Deep learning single models exhibited superior accuracy in detecting GAI compared to machine learning overall [57]. The authors of [58] performed the classification of Indonesian fake news using pre-trained models based on a multilingual transformer model (XLM-R and mBERT) combined with a BERTopic model as a topic distribution model. Their model provided an accuracy of 0.9051.

However, these models did not delve into the specifics of the generated news and the potential multi-class outcomes it could produce. Indeed, the previous researches did not establish the distinction between the different AI-generated classes, hence the need to generate a dataset containing the GAI-Fake and GAI-True classes. Effectiveness might arise from the intricate attributes of AI-generated content, subtle writing approaches, and the dynamic landscape of misinformation. Moreover, relying solely on single models [60, 61, 59] is proving insufficient. Addressing these limitations, more advanced and adaptable techniques, such as a dual-stream model, which is proposed in Section 4.3, could enhance accuracy and effectively counter the evolving challenges of fake news. Past research has

[7] the words True and Real will be used intrinsically. Likewise, for the words False and Fake.

predominantly relied on machine learning models or a single transformer model, except for the very recent paper [17], which proposed one dual model technique.

In order to classify and understand the nature of news (Fake, Real, GAI-True or GAI-Fake), one can check whether the news contains false, misleading or deliberately false information by: (1) ensuring the credibility of the news [64] even if it is generated by AI; and (2) using automatic detection tools [65] to distinguish authentic news from fake or AI-generated news.

2.3 Influence of generative AI on the proliferation of fake news

Generative AI is a sub-field of AI that focuses on developing models and algorithms capable of generating human-like content, such as text, images, and audio. Recently, text generation, in particular, has seen remarkable advancements, thanks to models like GPT-3 (Generative Pre-trained Transformer 3) [66] and its successor GPT-4 [67].

Generative AI has unfortunately been exploited for the generation and dissemination of deceptive news content. Ironically, advancements in technology aimed at detecting false information have accelerated the creation of increasingly sophisticated and perplexing fake news. The demarcation between reality and fabrication has become increasingly blurred with the advent of AI-powered text generation [68]. Indeed, the potency of generative AI possesses a dual nature: it enables the automatic generation of text closely resembling human writing, while, on the other hand, it significantly contributes to the generation of disinformation. In fact, by inputting a prompt or topic, these models can generate seemingly authentic news articles, reports, or social media posts that are entirely fictional [69, 70][8]. As per findings from researchers [9], the utilization of generative technology may render disinformation more cost-effective and simpler to generate, thereby contributing to an increase in the prevalence of conspiracy theorists and the dissemination of false information.

In a study conducted by researchers at OpenAI [71], a cautionary note was issued regarding the potential for the chatbot service to lower the costs associated with disinformation campaigns. It was suggested that malicious actors might be incentivized by the prospect of financial gain, advancing specific political agendas, or sowing discord and confusion. Furthermore, just two months after its launch, ChatGPT faced criticism from the NewsGuard platform [10], a specialized entity in the detection of fake news. NewsGuard noted that the chatbot had the potential to morph into a "superspreader" of disinformation. In an experiment detailed on their platform [11], the NewsGuard team instructed ChatGPT to compose articles that mirrored the viewpoints of prominent conspiracy theorists or biased news outlets. The results of the study uncovered that, out of the 100 misleading stories pre-identified, ChatGPT generated false news for 80 of them. These misleading narratives could appear compelling and authoritative to uninformed readers, despite their basis in falsehoods.

In response to this challenge, researchers have dedicated significant efforts to develop methods for detecting AI-generated content used for deceptive purposes. These methods encompass a variety of approaches, from employing machine learning models trained to recognize patterns in AI-generated text to linguistic analysis that seeks anomalies in language usage [72, 73, 74], including the architectures of transformers, which are elaborated in the subsequent section.

3 Background: transformers

Within this section, a brief overview of the architectural structures that underlie transformers will be provided.

3.1 Architecture

The transformer architecture, as elucidated in [75], adopts an encoder-decoder structure as shown in figure 1.

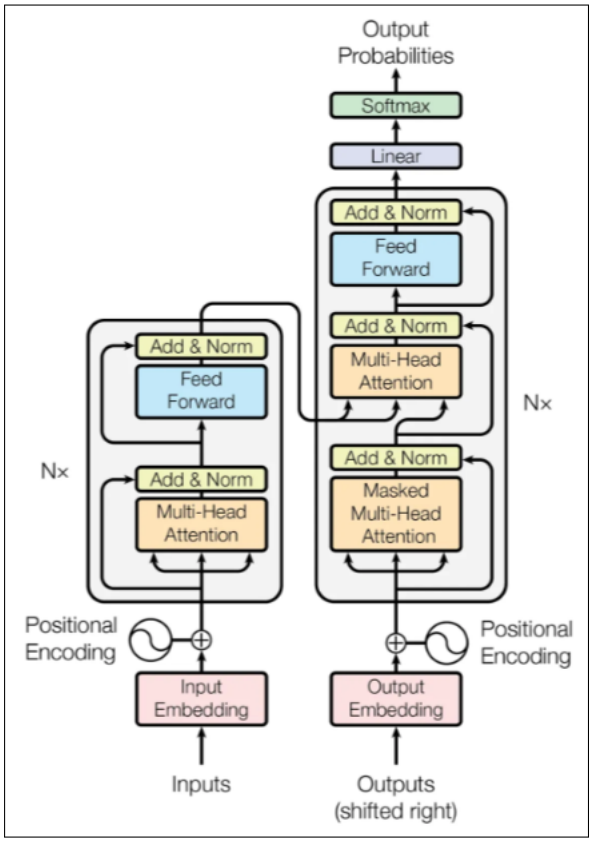


Figure 1: Original structure of transformer

[8] https://shorturl.at/boAHW, Last access: 21 August 2023
[9] https://shorturl.at/nKPS2, Last access: 21 August 2023
[10] https://www.newsguardtech.com/, Last access: 12 September 2023
[11] https://shorturl.at/auwTU, Last access: 12 September 2023

Transformer operates by taking an input sequence $X = (x_1, \ldots, x_N)$ and transforming it into a latent representation $Z = (z_1, \ldots, z_N)$. Notably, due to the auto-regressive nature of this model, the output sequence $Y_M = (y_1, \ldots, y_M)$ is generated one element at a time. In other words, each word y_M is generated using the latent representation Z and the previously created sequence $Y_{M-1} = (y_1, \ldots, y_{M-1})$.

Both the encoder and the decoder components of the transformer employ an identical multi-head attention layer, which plays a pivotal role in data processing. In this mechanism, a single Attention layer maps a query Q and keys K to a weighted sum of the values V. For practical reasons, a scaling factor of $\sqrt{\frac{1}{d_k}}$ is introduced to ensure effective operations:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

This attention mechanism is fundamental to the transformer's ability to capture complex relationships and dependencies in the input data, enabling its outstanding performance in various natural language processing tasks.

3.2 Transformers list

There are numerous transformer models developed for various natural language processing (NLP) tasks. Here are some notable ones:

- **Generative Pre-trained Transformer 2 (GPT-2):** a powerful natural language processing algorithm that can generate text that is both fluent and relevant to the context. It is used in a variety of NLP tasks, such as content creation, chatbots, question answering, sentiment analysis, and text summarization. GPT-2's ability to produce text that is similar to human-written text makes it a valuable tool for a wide range of language-related applications [76].
- **Bidirectional and Auto-Regressive Transformers (BART):** a sequence-generating paradigm that works well for producing text. In order to produce material that flows and makes sense, it can take into account both the context that comes before and after. BART is frequently utilized for machine translation, generative text synthesis, automated summarizing, and paraphrasing [77].
- **Bidirectional Encoder Representations from Transformers (BERT):** a powerful language model that can learn the meaning of words and sentences by reading them from both sides. This makes it very good at understanding the meaning of language and performing many different tasks, like classifying text, recognizing named entities, analyzing sentiment, and understanding context [78].
- **Distillable BERT (DistilBERT):** a smaller and faster BERT. It works just as well as BERT, but it uses less computing power and memory [79].
- **Robustly optimized BERT approach (RoBERTa):** a version of BERT that has been improved to perform better during pre-training. It is often used for the same tasks as BERT, but with better accuracy because of the changes made to its training process [80].

4 Proposed approach

In this section, we propose a three-part approach, namely creating an online survey, creating a new dataset, and exploring multiple dual models for classification and detection of fake news.

4.1 Online survey

We propose a survey with the comprehensive aim of investigating public perceptions and experiences related to fake news. The objective is to collect valuable insights into how individuals consume and distinguish information, the impact of fake news on their beliefs and decisions, and the measures they think can effectively combat misinformation.

The online survey was made available for participation from September 25th to November 25th, 2023. The questionnaire was distributed as widely as possible, and no filters were applied on participants in order to ensure variation in the target audience. Before completing the survey, participants were explicitly notified about the research nature of the survey. To evaluate a participant's existing knowledge without any external influence, the survey intentionally omitted an initial definition of technology. All responses were anonymized, and participants had the option to skip any question as none were obligatory. Participation was entirely voluntary, allowing individuals to exit the survey at their discretion.

To propose the questions for the survey, research results from [81, 82, 37, 25] were used. The selection of primary thematic domains, along with their associated inquiries, and all survey content and the data gathered during this research are openly accessible on https://rb.gy/kbt7lv.

4.1.1 Study design

The objective is to investigate the relationship between the five-factor model of personality, individual attitudes, and fake news detection ability. Drawing from the extensive literature review on socio-demographic, behavioral, and intercultural factors that impact the identification of fake news, this study organizes a series of sections [S1, S2,..., S5], each specifically designed to address corresponding Research Questions [S1Q1, S1Q2, ..., S2Q1, ...].

The survey encompassed a diverse range of question formats, including Yes/No, Likert scales, checkboxes, and multiple-choice responses. In questions using a Likert scale, participants were provided with either a 3-point scale ranging from 'agree' to 'disagree' or a 5-point scale spanning from 'strongly disagree' to 'strongly agree' (or 'not at all attentive' to 'extremely attentive', 'extremely disinterested' to 'extremely excited', 'never' to 'always', 'not familiar' to 'very familiar', 'unable' to 'able').

The survey comprised a total of 43 questions, organized into five distinct sections:

- **Demographic Information**: the first section (S1) included six questions [S1Q1-S1Q6] related to age, gender, country, level of education, current occupation, and income level.
- **Personality Traits**: comprising 14 questions [S2Q1-S2Q14], this section (S2) aimed to analyze how the respondent interacts with their environment, including behaviors, thoughts, and emotions.
- **Knowledge and Awareness**: summarized in 6 questions [S3Q1-S3Q6], this section (S3) assessed the respondent's understanding and consciousness of information, facts, or truths about a particular subject or their surroundings.
- **Attitudes Towards News and AI**: the fourth section (S4) contained nine questions [S4Q1-S4Q9] designed to detect the beliefs, perceptions, and stances individuals hold regarding the intersection of artificial intelligence and the consumption or dissemination of news and information.
- **Distinguishing Between Real and Fake News Generated by Artificial Intelligence**: this last section (S5) included six questions [S5Q1-S5Q6] aimed at measuring the respondent's ability to identify and discern the authenticity of news articles and information generated using AI technologies.

When designing a survey, it is imperative to establish a meaningful correlation between the different sections of the questionnaire. This correlation is a key determinant of how respondents are likely to respond to different parts of the survey. For this, aligning and interconnecting the sections of the survey facilitates a smoother flow of information and improves the overall understanding of respondents' views to provide a comprehensive view of respondents' capacity to detect fake news. For example, a section exploring demographic information may influence responses to subsequent sections focusing on opinions or behaviors [83]. Also, the personality traits section affects the decision of surveyors [84].

4.1.2 Data collection

The survey was made available on the Internet and distributed via various channels, including email, Facebook, and other social media platforms. The target audience was diverse, encompassing a wide range of age groups and educational backgrounds, and distributed across different geographical regions worldwide.

The study gathered a total of 103 responses. After excluding incomplete submissions, 83 valid responses were utilized for data analysis. The participant demographics encompassed a diverse range, including 1 individual below 18 years old, 40 participants aged 18 to 29, 33 between 30 and 49, and 8 between 50 and 64. The educational backgrounds of participants varied, spanning from primary education to higher education, and included a mix of unemployed individuals, working professionals, and those currently pursuing studies.

4.1.3 Evaluation method

Survey evaluation entails employing various methods to ensure the validity, reliability, and pertinence of the gathered measures. Among these techniques, structural equation modeling (SEM) emerges as a potent tool [87, 86, 85]. SEM aids in evaluating both convergent and discriminant validity of measures, offering insights into the precision of the survey's measurement instruments. Moreover, SEM enables the modeling of relationships between variables, thereby simplifying the analysis of the intricate structures inherent in surveys. In addition to SEM, other statistical evaluation approaches [12] [13] (mean, median, standard deviation, variance, ...) encompass reliability analysis to gauge the internal consistency of questions, factor analysis to unveil relationships between variables, and prior validation of the questionnaire by experts to ensure conceptual relevance.

SEM not only facilitates the measurement of correlations but also enables the exploration of causal relationships between survey sections. This capability aids in discerning whether one section of the survey significantly influences another. This provides a more deeper understanding of the dynamics between variables. Such insights serve to enhance the interpretation of survey results and offer more comprehensive understanding of the intricate interactions among the various dimensions. These collective methods fortify the validity of the data derived from the survey and make easier interpretation of the results. For the present study, SEM is predominantly employed in assessing the responses to the proposed survey.

4.2 New extended real and fake news (ERAF-News) dataset

The literature has introduced datasets for news that cover two or three classes: Real, Fake, and possibly AI-generated from real sources [88]. However, there is no distinction between AI-generated news from real or fake sources.
With the dataset ERAF-News, we aim to differentiate between these two new categories. In fact, distinguishing be-

[12] https://shorturl.at/pwzW8, Last access: 13 October 2023
[13] https://t.ly/fzrV8, Last access: 7 November 2023

tween real and fake news generated by AI offers several advantages, such as: (1) improving credibility by distinguishing between types of AI-generated information; (2) raising public awareness about the presence of AI-generated misinformation by verifying the accuracy of information regardless of its origin; (3) developing automated verification systems to improve algorithms and systems for authenticity and accuracy of online content; (4) informed decision-making based on accurate and correct data; and (5) limiting the spread of false information by identifying misleading content.
These objectives highlight the importance of developing tools and methods to analyze and evaluate AI-generated information. To this end, we propose to create our new dataset ERAF-News.

4.2.1 Building ERAF-News: methodology and process

Describing the methodology for dataset preparation is a pivotal phase in the research on fake news detection. To kick-start the research, the "Fake News Detection Datasets" from Kaggle was used [14]. This dataset encompasses two primary files: "True.csv", for authentic news, and "Fake.csv", containing fake news. These initial files consisted of roughly 23,000 and 21,000 entries, respectively, and included essential columns such as title, text, subject, and date.

The current study introduces a detection framework extending beyond the binary "True" and "Fake" classification. An additional category, "Generated", was incorporated specifically tailored for AI-generated news. This involved extracting data from "True.csv" and "Fake.csv" to create two new sub-datasets named "GAI-X-True.csv" and "GAI-X-Fake.csv" where X denotes the name of the model used for the generation. Consequently, six distinct dataset files were maintained, each representing a unique news category. The objective behind this expanded categorization is to provide a more nuanced and precise foundation for classification.

Figure 2 illustrates the process of obtaining the generated ERAF-News dataset. For the generation process, the choice from the introduced models in section 3.2 was guided by considerations mentioned from [89]. They showed that GPT, BERT, and BART are versatile, capable of both generation and classification tasks, and have demonstrated commendable accuracy.

4.2.2 Generation techniques

The creation of the generated dataset involved a series of generation experiments executed on the Google Colab platform. Pre-trained models were employed to process the original text and produce the corresponding segments of what is now referred to as the "Extended Real And Fake News Dataset" (ERAF-News dataset). This comprehensive dataset offers a valuable resource for researchers and developers looking to advance AI-generated fake news detection. We've made an extract of ERAF-News dataset freely available for unrestricted use on shared drive available at the link https://rb.gy/kbt7lv in order to facilitate collaboration and further research in this critical domain.

From "True.csv" and "Fake.csv", 15000 entries were randomly selected. These entries were used as inputs for generating new classes using the BART model, resulting in "GAI-BART-True.csv" and "GAI-BART-Fake.csv". Likewise, the same set of inputs was used for the GPT-2 model, yielding "GAI-GPT2-True.csv" and "GAI-GPT2-Fake.csv". Concluding the process, upon applying BERT to identical inputs, "GAI-BERT-True.csv" and "GAI-BERT-Fake.csv" are generated. The performance metrics used to evaluate the generation process are discussed in section 5.2.

This dataset was created specifically to be used in the novel classifier architecture, as discussed in the next section.

4.3 Proposed dual-stream transformers model (DuSTraMo)

In this section, we present a pioneering strategy to address the issue of news classification, introducing an innovative dual-stream deep learning framework. Given the escalating prevalence of fake news and disinformation, it's imperative to have advanced solutions that can swiftly and precisely determine the accuracy of news articles. We propose an approach based on a dual-stream architecture illustrated in figure 3.

Each stream in the dual-stream model is assigned a pre-trained deep learning model. The outputs of these two streams are subsequently combined through concatenation, enabling the global model to efficiently learn the cross-modal interactions and synergies between the viewpoints of the two models. This fusion of information produces more complete and contextually rich representations, often leading to improved performance in downstream tasks. Fine-tuning with additional dense layers further tailors the entire model to the specific target task, creating a versatile framework adaptable to various classification tasks.
The dual-stream model enhances the learning of complex patterns and relationships, thereby improving the overall accuracy and resilience of classification by enriching feature representation through this architecture [91, 90]. Com-

[14] https://rb.gy/d5mhig, Last access: 25 March 2023

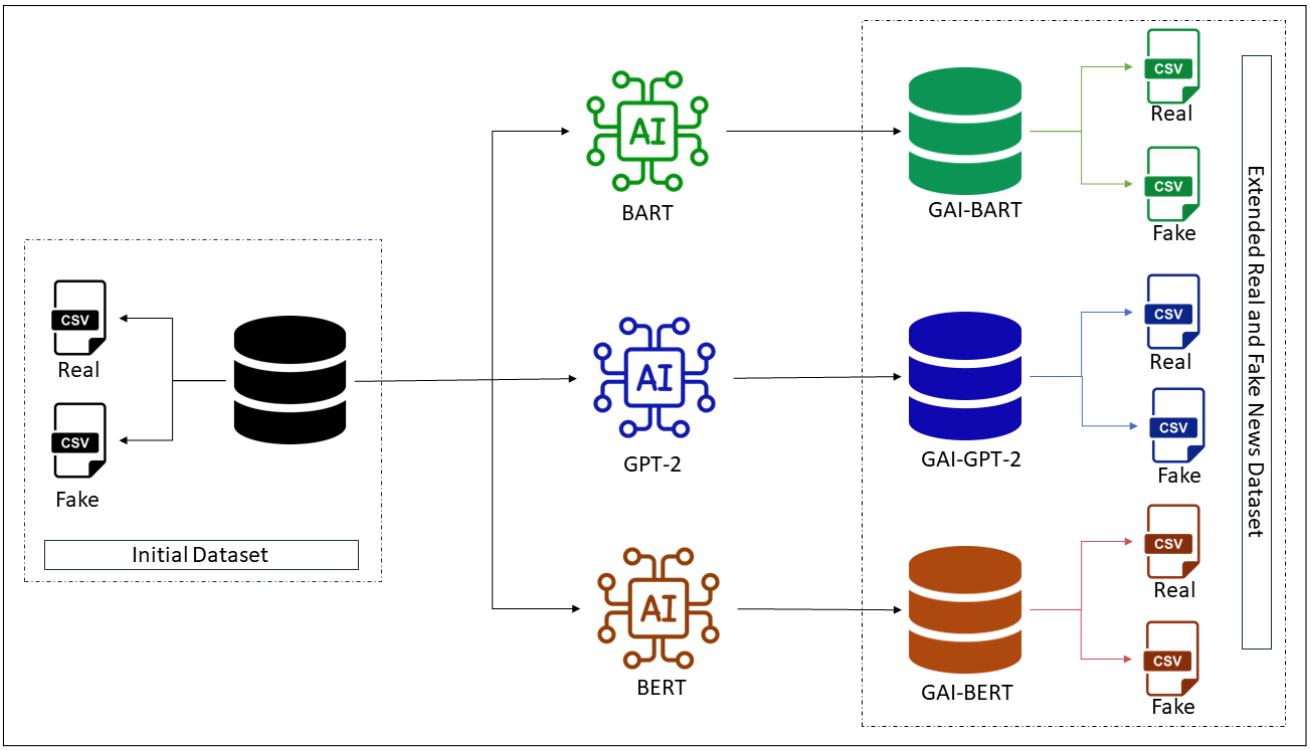


Figure 2: Extended real and fake news (ERAF-News) dataset

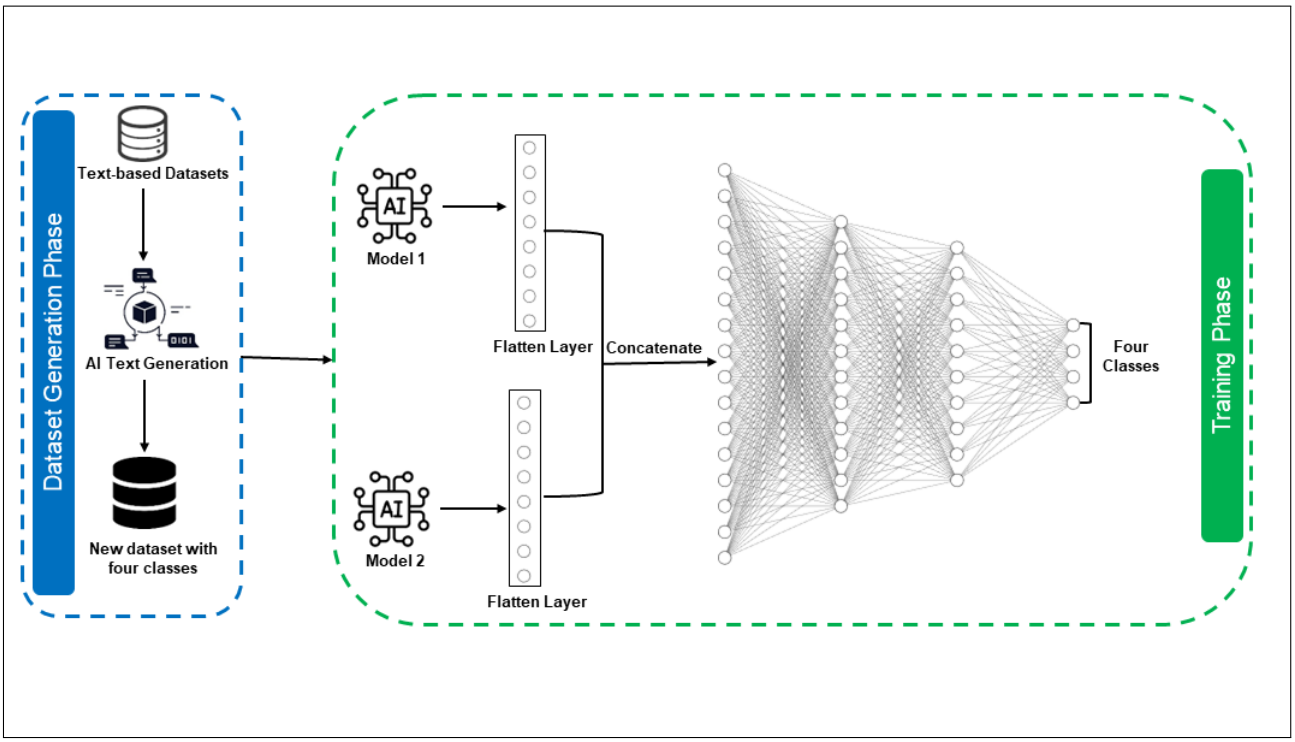


Figure 3: Dual-Stream transformers model (DuSTraMo)

pared to single-model techniques, dual models offer several advantages. Firstly, they perform exceptionally well by combining the strengths of two specialized models, resulting in increased accuracy and improved generalization. Secondly, the synergy between dual models allows for the capture of multiple facets of the data, potentially leading to more accurate predictions. Ultimately, by presenting multiple perspectives on the data, this model has the potential to enhance the explainability of the obtained results [94, 93, 92]

For the classification tasks, the choice among the models cited in section 3.2 was guided by [59, 95] which involved a comparative analysis of the BERT, BART, GPT-2, RoBERTa and DistilBERT models, revealing the competence of these models.

In this paper, we emphasize flexibility in the dual-stream architecture. At any given instance, the first stream could be a GPT-2, BART, or BERT model. The second stream introduces variability, which could consist of the same model as the first or alternatively, DistilBERT or RoBERTa. This yields three distinct configurations:

- The GPT-2-based configuration includes dual-GPT-RoBERTa, dual-GPT-DistilBERT, and dual-GPT-GPT combinations.
- The BART model-based configuration comprises dual-BART-RoBERTa, dual-BART-DistilBERT, and dual-BART-BART.
- The BERT model-based configuration encompasses dual-BERT-RoBERTa, dual-BERT-DistilBERT, and dual-BERT-BERT combinations.

In summary, the three models GPT-2, BART, and BERT serve as constants in the first stream of the dual model due to their roles in the generation process. Additional models, such as RoBERTa and DistilBERT, along with duplicated first stream models, are incorporated into the second stream, showcasing notable classification capabilities as evidenced by previous literature.

To the best of our knowledge, the architecture of a dual-stream classifier has been introduced in the fake news domain only twice: first in [57], which employed simpler (Machine Learning) streams (such as SVM, LSM, RNN, ...), and more recently in [17], which used BERT in both streams. In contrast, the present paper proposes a novel approach by utilizing diverse streams based entirely on transformers.

5 Findings

5.1 Assessment of survey results

In this section, the study and analysis of the survey results are presented using the SEM methodologies. The survey collected responses from 103 participants, 83 of which were valid, spanning 9 countries around the world. The subsequent analysis is conducted on individual sections of the survey, followed by an examination of the interplay and mutual influences among them.

5.1.1 Survey evaluation metrics

The SEM method can include several metrics, depending on its configuration and the specifications of the structural model [96]. Commonly used metrics in SEM include:

- **Chi-square (χ^2):** evaluates the fit of the model to the observed data. A low chi-square indicates a good fit.
- **Degrees of Freedom:** represents the maximum number of logically independent values.

Other metrics commonly used in the fields of statistics, research, and data analysis are:

- **Conditional Demographic Disparity (CDD):** assesses whether a facet exhibits a higher proportion of rejected outcomes in the dataset compared to accepted outcomes.
- **p-value:** indicates the probability that the observed results in statistical analysis are due to chance. In the context of hypothesis testing, a low p-value (typically < 0.05) suggests that the results are statistically significant, providing evidence to reject the null hypothesis.
- **Coefficient of Determination (R^2):** indicates how well the independent variables can explain the variation in the dependent variable.
- **Completion Rate:** measures the proportion of people who answered all the questions relative to the total number of people invited or contacted. It is an indicator of the survey's engagement and effectiveness.
- **Correlation:** measures the statistical relationship between variables. It can be positive (variables move in the same direction), negative (variables move in opposite directions), or null (no linear relationship). It quantifies the strength and direction of this relationship.

In addition, various metrics can be used to measure different characteristics or aspects namely: frequency, percentage, mean, standard deviation, mean score, etc.

5.1.2 Demograhic evaluation

Conducting a demographic assessment in survey research is crucial for gaining a comprehensive understanding of the surveyed sample. Table 3 resumes the participants' data according to the demographic section.

Figure 4 illustrates the distribution of survey participants based on their respective countries.

Conducting a comparative analysis of information regarding "country residence" and "income level," the following outcomes were obtained:

- The Cramer Dependence Coefficient (CDD) registered a value of 35.22%, signifying a substantial correlation between the "country" and "income level" variables. This CDD value points to a notable association, implying that the income level is closely linked with the respondent's country of residence.

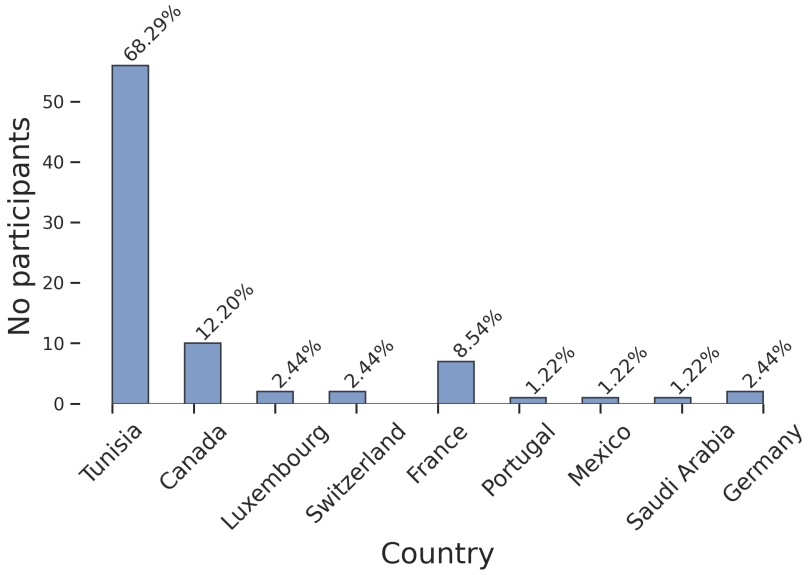


Figure 4: Countries participation histogram

Table 3: Participants demographic data

Variable	Values	N (%)
Gender	Male	44 (42.7%)
	Female	59 (57.3%)
Age	Under 18	1 (1%)
	18-29	50 (48.5%)
	30-49	42 (40.8%)
	50-64	10 (9.7%)
	65 or older	0
Current occupation	Student	45 (44.1%)
	Worker	56 (54.9%)
	Unemployed	1 (1%)
Education Level	Secondary Education	2 (2%)
	Bachelor's degree	13 (13%)
	High school or equivalent	16 (16%)
	Master's degree	46 (46%)
	Doctoral degree	20 (20%)
	Prefer not to answer	3 (3%)
Income Level	Low	20 (19.6%)
	Medium	45 (44.1%)
	High	13 (12.7%)
	Prefer not to answer	24 (23.5%)

– The Chi-square test statistic yielded a relatively high result of 30.52%, measuring the disparity between observed data and anticipated values under the null hypothesis, which assumes independence between the variables "country" and "income level." This outcome underscores a significant association between the two variables.

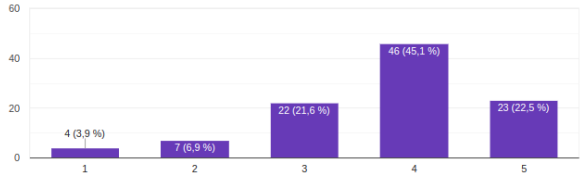
– The associated p-value, measuring the likelihood of obtaining results as extreme as those observed, was calculated as 16.81% in the context of the Chi-square test. Although this value exceeds the commonly used significance threshold of 5%, indicating insufficient evidence to reject the null hypothesis of independence, it is noteworthy that the p-value is not extremely high, suggesting a tendency towards association despite non-rejection.

– A total of 24 degrees of freedom were observed, a parameter contingent on the size of the contingency table. This substantial degree of freedom, reflective of the relatively large dataset, reinforces the statistical robustness of the analysis. In summary, these findings underscore a moderate to a strong association between "country of residence" and "income level", and they affirm the adequacy and reliability of the analyzed sample comprising 83 valid survey responses.

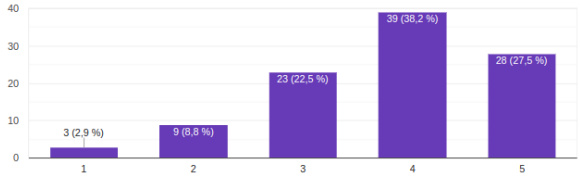
Similarly, an analysis of "age" and "current occupation" reveals a strong and statistically significant association, as evidenced by a CDD of 47.80%, a high chi-square test value of 37.47%, and an extremely low p-value 1.42e-06 with an interaction degree of freedom of 6.

5.1.3 Personality traits evaluation

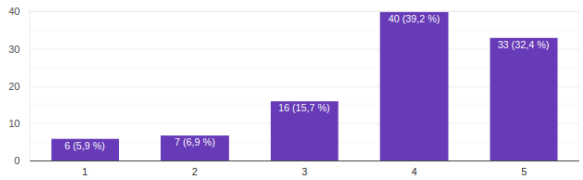
The second section (named S2) of the questionnaire delves into personality traits. As illustrated in figure 5, it's evident that a significant portion of individuals who value organization and tidiness (44.3%) also tend to engage in advance planning (38.1%) and exhibit enthusiasm for exploring new experiences (39.2%). These characteristics notably impact an individual's disposition during conversations, with 58.3% reporting a high level of comfort.



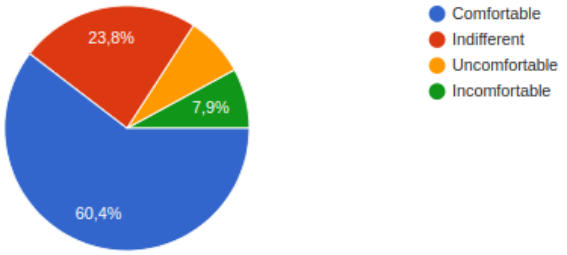
(a) Having an organized workspace.



(b) Tend to plan activities in advance.



(c) Ability to try new experiences.



(d) Feeling in initiating conversations.

Figure 5: Global personality traits analysis

Figure 6 summarises an assessment using the Big Five method produced mean values for Openness in [0-2], Conscientiousness in [0-3], Extroversion in [0-3], Agreeableness in [0-2] and Neuroticism in [0-2]. Specifically, the obtained results revealed mean and standard deviations for the Big five personality factors as follows: openness (M = 1.21; SD = 0.52), extroversion (M = 1.21; SD = 0.51), agreeableness (M = 1.19; SD = 0.53), neuroticism (M = 0.75; SD = 0.40), and conscientiousness (M = 1.64; SD = 0.53).

5.1.4 Knowledge and awareness evaluation

Analyzing the results of the third section (S3), the survey participants indicated that many of them rely on international TV for their news, and several of them do not actively check its accuracy. However, most of them have limited knowledge about concepts such as fake news, and AI, and are unfamiliar with the term ”ChatGPT”. This is due to a lack of awareness about GAI.

Figure 7 illustrates the breakdown of survey participants' responses regarding their level of knowledge and awareness.

As outlined in section 4.1.1, the third section of the survey (S3) comprises eight questions (S3Q1-S3Q8). The distribution of responses has been assessed for each question. The responses have undergone pre-processing and normalization, resulting in values within the 0 to 1 range. The initial histogram illustrates the distribution of responses to the first question (S3Q1), with the predominant choice being the second answer. Similarly, for the histogram corresponding to S3Q2, the majority of participants selected the third answer, and this pattern persists across the subsequent histograms.

5.1.5 Attitudes towards news and AI evaluation

This section explores respondents' perspectives on fake news, including their ability to detect it, their capacity for verification, and their potential for guarding against unintended consequences of AI-generated fake news. Following data pre-processing, figure 8 summarizes the calculated attitude scores.

As this section comprises nine questions with scores that can range from 0 to 9, it is noteworthy that 54.90% of respondents attained scores between 4 and 6 (figure 8a). This observation suggests that a significant portion of the participants exhibit psychological equilibrium in their attitudes toward fake news. Furthermore, a p-value was calculated to assess the relationship between the respondent's country of residence and their attitude scores. The p-value, approximately 7.82%, suggests that there are statistically significant differences in attitude scores across the various countries. Even though 57.31% of the respondents rely on the Internet for 75% of their information, it is noteworthy that 47.56% of them expressed uncertainty about their ability to discern fake news from headlines alone (figure 8b).

It can be additionally observed that the practice of utilizing the Internet to acquire information is associated with the capacity to discern fake news. Specifically, as shown in figure 9, the linear correlation matrix between the variables of this section and its general score reveals that question S4Q2 (”about the use of the Internet as a source of information”), exerts the most significant variance on this score, accounting for 50%.

5.1.6 Distinguishing between real and fake news generated by AI evaluation

The fifth section (S5) contains six questions [S5Q1-S5Q6], with score means that ranged from 0 to 6. The survey results

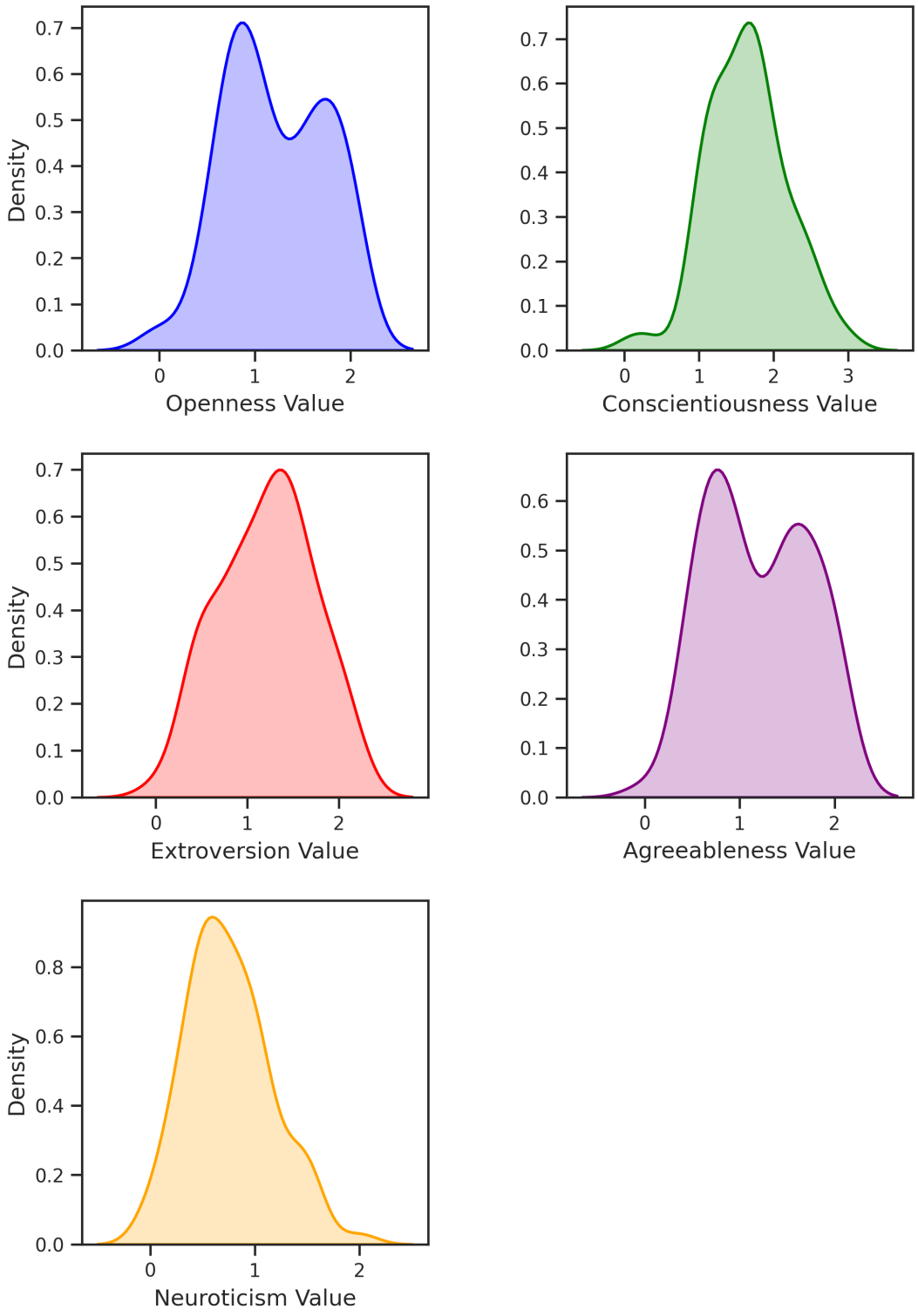


Figure 6: Assessment curves of participants' personality traits

showed an average score of 2.73, indicating that they are only able to distinguish between fake and real news to a degree of 40.19%.

Additionally, a computation of the correlation matrix, as depicted in figure 10, reveals the highest correlation at approximately 45%. Such a correlation is typically categorized as moderate. This suggests that there exists an average relationship between questions S5Q1 and S5Q2.

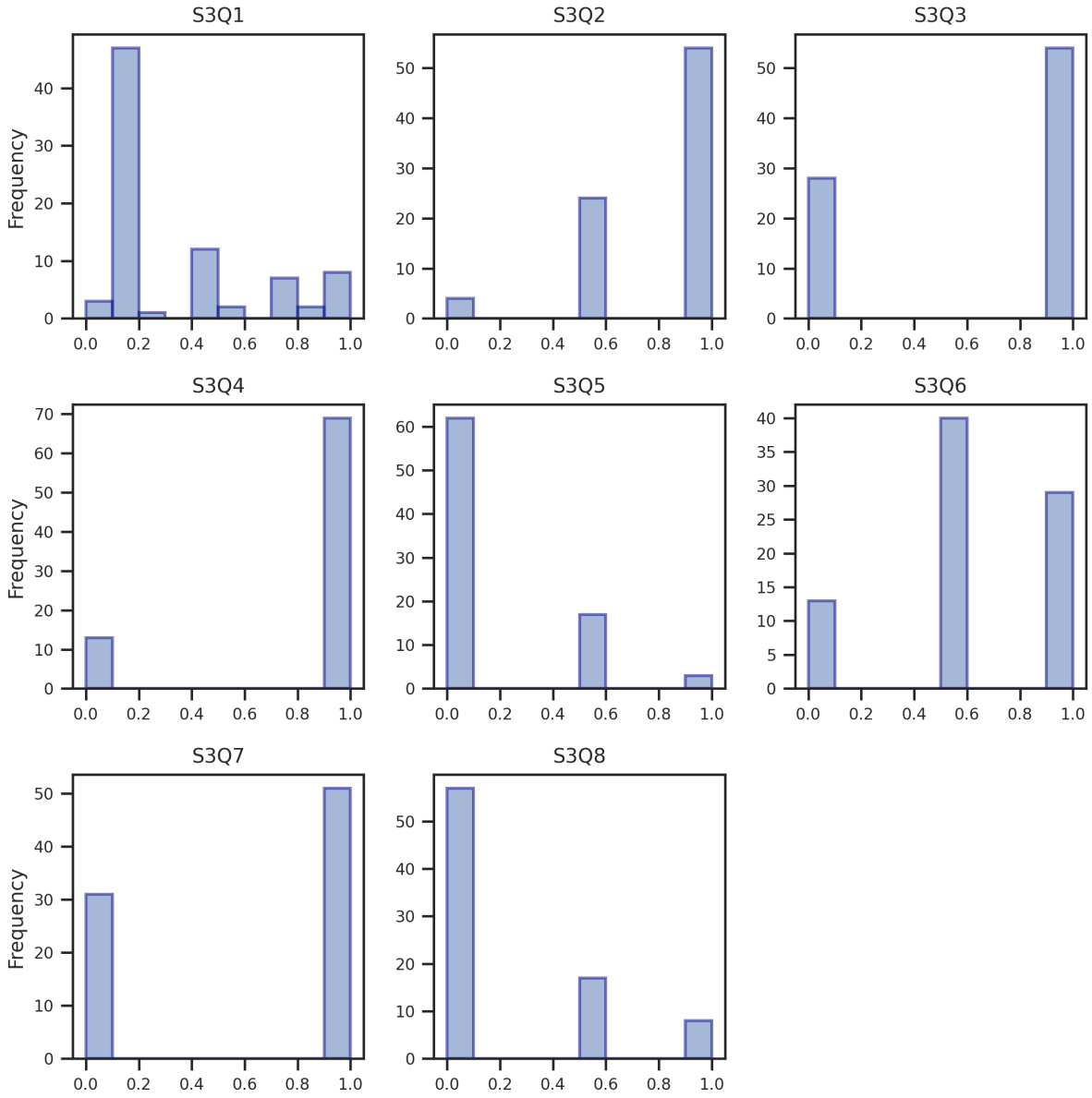


Figure 7: Distribution of knowledge level and awareness

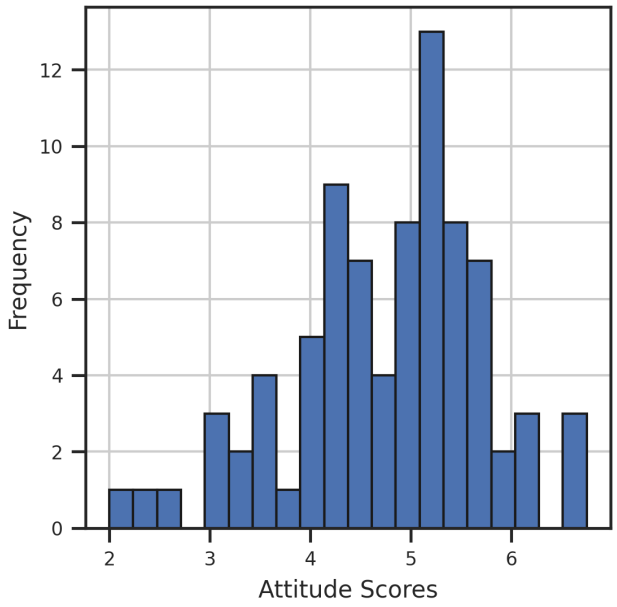
5.1.7 Survey summary

A total of 103 survey participants engaged in the study through an internationally published online questionnaire over 3 months. The uniqueness of this investigation resides in its examination of individuals' capacity to discern fake news based on the influence of demographic characteristics and personality traits according to the Big five personality model (OCEAN). In figure 11, a correlation analysis was performed using the five personality traits from the Big Five section (S2) to demonstrate their relationship on the sections S3 ('Knowledge and Awareness'), S4 ('Attitudes Towards News and AI'), and S5 ('Distinguishing Between Real and Fake News Generated by AI').

The findings reveal that:

- Section S3 ("Knowledge and Awareness") is most notably correlated with openness (61%) and extroversion (49%).
- Conscientiousness exerts a 43% correlation on Section S4 ("Attitudes Towards News and AI").
- Conversely, Section S5 ("Distinguishing Between Real and Fake News Generated by AI") is primarily explained by the agreeableness characteristic, contributing to 68% of its influence."

The results indicate that the percentages of intersecting influences are not exceedingly high, hovering around 50%. While the evaluations did underscore distinctions within



(a) Number of responses by attitude score

S4Q8 / S4Q2	0	1	2	3	4	Total
1	0	1	1	1	1	4
2	4	2	6	1	1	14
3	5	2	8	1	1	17
4	15	3	10	2	0	30
5	7	0	7	1	2	17
Total	31	8	32	6	5	82

(b) Cross table between habit of using the internet to gather information and ability to identify fake news

Figure 8: Evaluation of the ”Attitudes Towards News and AI Evaluation”

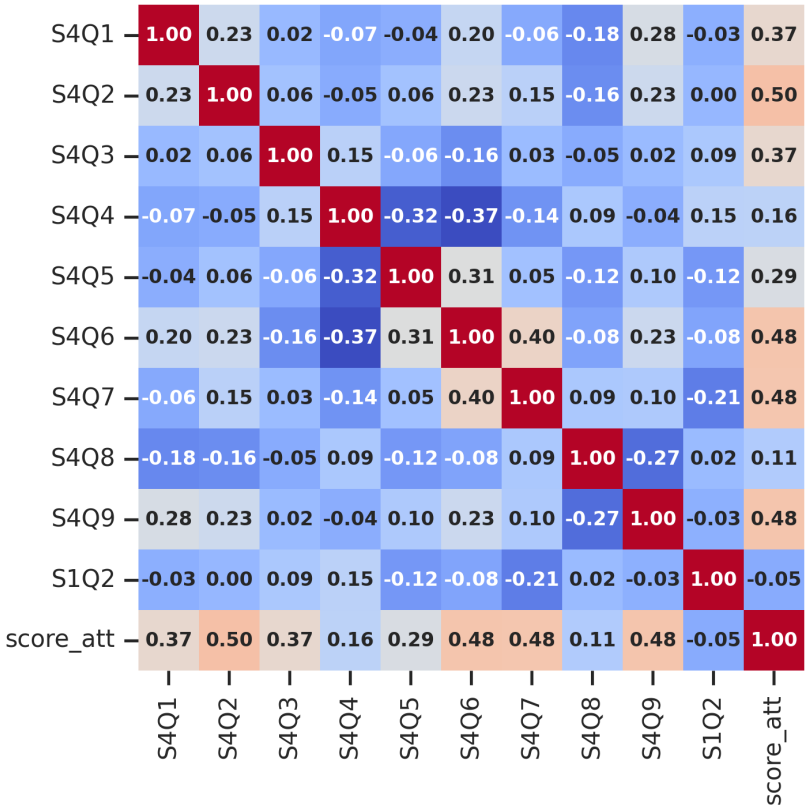


Figure 9: Correlation of Internet use on the attitude of responders

survey results, they nonetheless proved inadequate to attain a robust level of performance. Consequently, the study grapples with certain limitations. Firstly, the study's outcomes were derived from a relatively modest cohort of 83 users possessing personalities of moderately acceptable openness, conscientiousness, and extroversion.

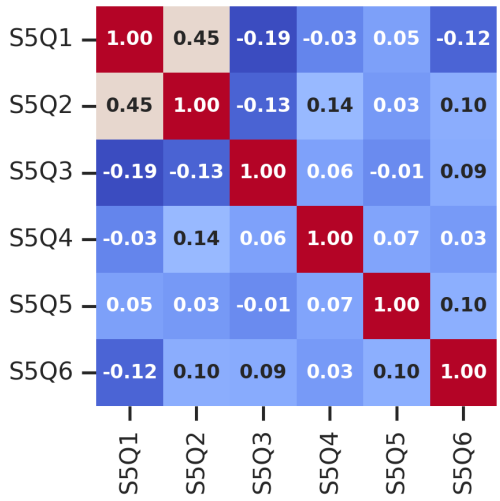


Figure 10: Correlation matrix for distinguishing between GAI real and fake news

5.1.8 Discussion and limits of surveying method

The outcomes of the survey indicate that only a small proportion of participants potentially can discern between authentic and fabricated news, with the majority appearing to lack this skill. This finding accentuates the necessity for an automated method to detect fake news, aimed at assisting users in identifying misinformation.

In this context, the application of AI serves as a crucial tool to enhance the effectiveness of these detection strategies. To ensure improvement in results through AI techniques, a study was conducted on survey questions. Specifically, we tested GPT, BART, and BERT models to predict responses. Table 4 showcases the results of the predictions made.

Table 4: Assessment of survey question prediction

	GPT	BART	BERT	DistilBERT	RoBERTa
GAI-GPT	**0.82**	0.71	0.77	0.50	0.25
GAI-BART	0.45	0.62	0.68	0.33	0.17
GAI-BERT	0.67	0.33	0.50	0.50	0.50

It was noted in section 5.1.6 that survey respondents can correctly answer questions from Section S5 with a percentage of 40.19%. However, with the integration of AI prediction, it has been demonstrated that we can achieve even a percentage of 82% of accuracy.

As we progress with this research, we strongly advocate for the integration of AI, which promises to introduce new, more robust dimensions in the realm of fake news detection. This forward-thinking approach is expected to surmount existing challenges by harnessing the power of machine learning and sophisticated data analysis, thereby revolutionizing the current landscape of misinformation identification.

The next section discusses an automated system grounded in artificial intelligence to surmount identified challenges and markedly enhance the performance of incorrect information detection.

5.2 Assessment of data generation quality

As previously mentioned, the ERAF-News dataset serves as the data source, comprising three distinct generated setsusing pre-trained models: BART, BERT, and GPT-2. The generated output deserves to be evaluated to measure the degrees of similarity and fluency of the generated text compared to the original text.

In the following, a list of metrics used are defined, then an evaluation of the generation quality of the ERAF-News dataset is presented.

5.2.1 Generation evaluation metrics

Metrics play a crucial role in evaluating results following text generation. The most commonly used metrics in the generation context are:

1. **BLEU score** (Bilingual Evaluation Understudy) focused on precision, initially used for translation but can be leveraged for generation evaluation [15]. BLEU score typically ranges from 0 to 1, where a score closer to 1 indicates a higher similarity between the generated text and the reference texts.

2. **ROUGE score** (Recall-Oriented Understudy for Gisting Evaluation) emphasizes recall, is Based on n-gram overlap [16]. Higher ROUGE scores indicate better performance in terms of matching the generated text to the reference texts.

3. **BERT score** is a metric designed to evaluate machine translation. It calculates a similarity score between each token in the candidate sentence and each token in the reference sentence. This is achieved by utilizing contextual embeddings from pre-trained BERT models and comparing words in candidate and reference sentences using cosine similarity. Additionally, BERT score provides valuable insights for the evaluation of diverse language generation tasks [97].

4. **BLEURT score** (Bilingual Evaluation Understudy with Representations from Transformers) is a metric designed for evaluating the quality of machine-generated text [98]. It focuses on evaluating the fluency and adequacy of generated text.

[15] http://tinyurl.com/2p96r5bx, Last access: 23 August 2023
[16] https://n9.cl/17pto, Last access: 23 August 2023

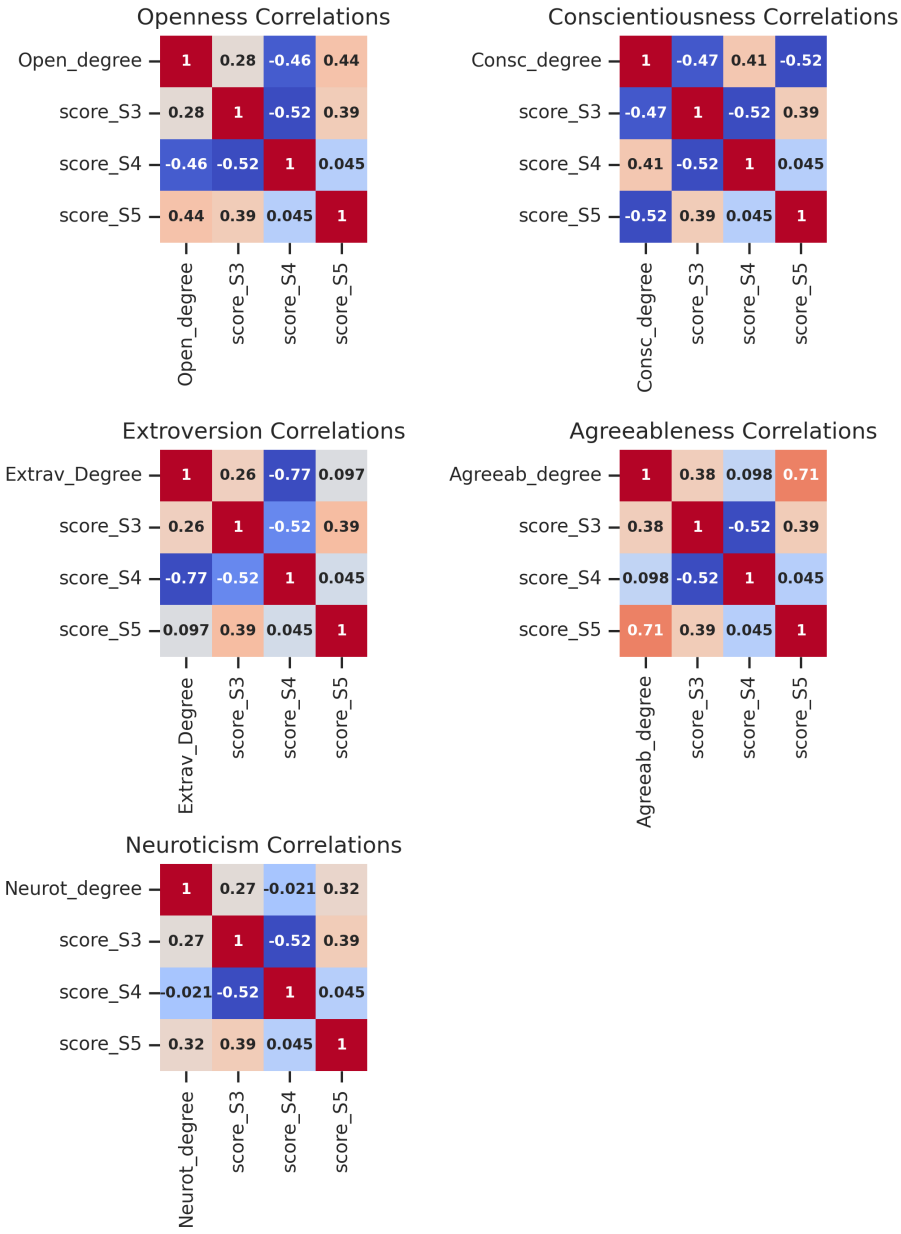


Figure 11: Influence of big five traits on other sections

5.2.2 Generation results

In Table 5, the evaluation of the generated dataset was conducted using various metrics and compared to prior studies. This assessment aimed to measure how closely the generated dataset resembles the original one.

The results of BLEU and ROUGE scores varied in the literature between [0.089-0.5] and [0.1-1.2] respectively. The multi-class classification carried out on ERAF-News exceeds these results and shows superior performances of the order of [0.203-0.256] and [0.366 - 0.916] respectively. In terms of BERT and BLEURT scores which varied between [0.2-0.6] and [0.3-0.7] respectively in the literature, reach overall scores higher than 0.91 and 0.85 respectively when carried out on ERAF-News dataset. As a conclusion, among the evaluated metrics, BART consistently emerges as the top-performing model, excelling in BLEU, ROUGE, BERT scores, and BLEURT scores. Its comprehensive success across multiple evaluation criteria positions BART as the most robust and effective model for text generation tasks. Additionally, BERT demonstrates strong performance across various metrics, showcasing its versatility in generating text that aligns closely with reference material.

5.3 Assessment of proposed method

This section provides a detailed exploration of the experiments conducted and a thorough analysis of the results ob-

Table 5: Comparative study of generation results

Year	Ref.	Dataset	Model	BLEU	Rouge	BERT score	BLEURT score
2019	[89]	WMT'16 ELI5	BART	0.379	1.2	-	-
2020	[97]	WMT18	BERT	0.527	0.536	0.693	-
	[98]	WMT17-19	BERT	0.2-0.3	-	0.25-0.45	0.3-0.57
2021	[99]	CovidDialog	GPT2	0.094	-	-	-
			BART	0.089	-	-	-
2022	[100]	WMT19	BART	0.206	0.165	0.317	0.325
	[101]	WebText CNNDM	NLP models	-	0.286	0.332	0.716
2023	current study	ERAF-News	GAI-GPT	0.203	0.366	0.917	0.853
			GAI-BART	0.216	0.383	0.957	0.983
			GAI-BERT	**0.256**	**0.916**	**0.918**	**0.929**

tained from the proposed method. The evaluation begins by assessing the individual performance of the selected models, followed by evaluating the dual models using various combinations.

5.3.1 Experimental setup

Single model executions were conducted on Kaggle, using a GPU T4x2 accelerator. On the other hand, for the dual model executions, Google Colab Pro and Kaggle platforms were used, leveraging GPU processing.

Several important parameters were used during the model's execution phase to guarantee efficient training and assessment. The dataset is divided into a training set comprising 80% of the data, and a test set comprising the remaining 20% of the data. The training process utilized a batch size of 2 for both single and dual models, with model weights adjusted over 2 epochs for each. Additionally, training performance was monitored and hyper-parameter adjustments are made using a validation split of 20% of the training set.

5.3.2 Used metrics

Evaluating the performance of various models includes the examination of detection metrics such as accuracy, precision, recall, and F1-score [103, 102]:

- **Accuracy:** the percentage of all correctly identified data points and the percentage of all correctly predicted data points (both positive and negative) are the same.
- **Precision:** the ratio of true positives to all real positive instances is the same as the ratio of true positives to all positive predictions.
- **Recall:** the proportion of true positive predictions among all actual positive cases is the same as the proportion of true positives among all positive predictions.
- **F1 score:** F1 score is calculated by taking the harmonic mean of the model's precision and recall.

These metrics are commonly used to measure the performance of machine and deep learning models, particularly in classification tasks.

5.3.3 Results of single models multi-classification

Table 6 displays the performance metrics of various single models multi-classification on different datasets: GAI-GPT2, GAI-BERT, and GAI-BART along with a comparison to models documented in the literature.

In the literature, previous works with single transformer models for classification predominantly utilized models such as RoBERTa, BERT, and XLNet. Their evaluations were primarily based on Accuracy and F1-score, ranging between 0.7 and 0.9. In contrast, the present study undertakes the RoBERTa and BERT models by incorporating additional models such as GPT-2, BART and DistilBERT and includes measures of precision and recall scores.

Multi-class classification with RoBERTa, regardless of the dataset, emerged with the lowest performance, with an accuracy of 0.2 and all other metrics being null, demonstrating its incapacity to multi-classify any type of generation. Similarly, BART showed very weak metrics, around 0.2, indicating its inability to perform multi-classification for any type of generation. In contrast, the BERT and DistilBERT models achieved perfect metrics, indicating that they made major predictions correctly. GPT-2 obtained equivalent values for all metrics, around 0.7, reflecting relatively consistent performance. Finally, GPT-2 demonstrates a balance between the values of its metrics but with lower performance compared to BERT and DistilBERT.

In summary, DistilBERT demonstrates excellent classification capabilities of all datasets, making it a promising choice for applications requiring high precision in class predictions.

Table 6: Comparative study of single models results

Year	Ref.	Dataset Used		Model	Accuracy	Precision	Recall	F1 score
2021	[104]	PolitiFact		RoBERTa	0.825	-	-	0.805
				BERT	0.88	-	-	0.87
				XLNet	0.895	-	-	0.90
		GossipCop		RoBERTa	0.803	-	-	0.807
				BERT	0.85	-	-	0.79
				XLNet	0.855	-	-	0.78
2022	[95]	LIAR		BERT	0.77	-	-	-
2023	[105]	real-world		BERT	0.8843	0.8937	0.8756	0.8846
	[72]	In-House		BERT	0.919	-	-	-
				RoBERTa	0.961	-	-	-
		TweepFake		BERT	0.891	-	-	-
				RoBERTa	0.911	-	-	-
	current study	ERAF-News	GAI-GPT2	RoBERTa	0.2530	0.0000	0.0000	0.0000
				BERT	**0.9787**	**0.9787**	**0.9787**	**0.9770**
				GPT2	0.7245	0.7241	0.7241	0.6348
				DistilBERT	**0.9845**	**0.9845**	**0.9845**	**0.9833**
				BART	0.2463	0.2460	0.2460	0.0973
			GAI-BART	RoBERTa	0.2467	0.0000	0.0000	0.0000
				BERT	**0.9737**	**0.9722**	**0.9721**	**0.9760**
				GPT2	0.6065	0.7460	0.4262	0.4729
				DistilBERT	**0.9643**	**0.9647**	**0.9638**	**0.9604**
				BART	0.2468	0.2468	0.2468	0.0977
			GAI-BERT	RoBERTa	0.2412	0.0000	0.0000	0.0000
				BERT	**0.9775**	**0.9775**	**0.9775**	**0.9759**
				GPT2	0.8398	0.8403	0.8388	0.8287
				DistilBERT	**0.9720**	**0.9720**	**0.9720**	**0.9694**
				BART	0.2538	0.2533	0.2533	0.0993

5.3.4 Single models discussion

Performance of models varies significantly across datasets, and each one exhibits distinct strengths and weaknesses. DistilBERT and BERT consistently delivered the best results across datasets, showcasing high precision, recall, and F1-scores. On the other hand, BART RoBERTa demonstrated inferior performance with very low Accuracy. GPT-2 model showed reasonably good performance, although with variations across datasets. These findings highlight the nuanced performance characteristics of each model, emphasizing the importance of considering both effectiveness and efficiency in choosing a model for specific applications.

5.3.5 Results of dual models multi-classification

Table 7 offers a comprehensive breakdown of the performance metrics for each dual model across various sub-datasets. The table displays four key evaluation metrics: accuracy, precision, recall, and F1 score.

The literature on dual models for fake news classification is relatively limited. Previous works, as mentioned in [57], incorporated dual-ML models to achieve accuracy ranging between 0.92 and 0.95. The most recent research [17] utilized a dual-BERT-BERT model, achieving an accuracy of 0.85.

Applying DuSTraMo on ERAF-News yielded results in 2 classes: (1) some dual models' highly performing outcomes exceeding 0.95 of accuracy, surpassing the literature's benchmarks; and (2) other dual models showing instability in their results. Sometimes, they provide extremely poor results or very good ones. And at other times they simply did not work at all, especially with dual-BART-BART (indicated as NE for non-executable). In addition, it was very challenging to execute due to their enormous dimension and memory requirements.

For more details, the GPT dual model applied to the GAI-GPT2-generated dataset consistently achieved the remarkably high accuracy score of 0.9999. This observation underscores the model's exceptional performance when applied to datasets intended for GPT2 generation. Conversely, for the GAI-BART dataset, the dual-GPT2-DistilBERT model outperformed other model combinations with an accuracy of 0.9885. Finaly, for the GAI-BERT generated dataset, the BERT-RoBERTa dual model outperformed other model combinations with an accuracy of 0.9881.

5.3.6 Dual models discussion

Present research meticulously explores a range of configurations, employing diverse models to ascertain their ef-

Table 7: Results of DuSTraMo models

Year	Ref.	Dataset Used		Dual Models		Acc.	Prec.	Recall	F1 score
				Model 1	Model 2				
2022	[57]	CONSTRAINT shared task-2021		BERT-XLNet-ELMo		0.93	-	-	0.925
				LSTM-BiLSTM GRU-BiGRU		0.92	-	-	0.925
				LR-SVM-RF-KNN BERT		0.95	-	-	0.95
2023	[17]	FakeNewsNet		BERT	BERT	0.854	0.756	0.555	0.640
2024	Current study	ERAF-News	GAI-GPT2	GPT2	GPT2	**0.9999**	**0.9999**	**0.9999**	**0.9998**
					RoBERTa	0.9995	0.9995	0.9995	0.9995
					DistilBERT	0.9998	0.9998	0.9998	0.9997
				BART	BART	NE	NE	NE	NE
					RoBERTa	0.9973	0.9973	0.9973	0.9971
					DistilBERT	0.9977	0.9977	0.9977	0.9977
				BERT	BERT	0.3239	0.9208	0.0729	0.1044
					RoBERTa	0.9670	0.9670	0.9668	0.9651
					DistilBERT	0.9833	0.9833	0.9833	0.9826
			GAI-BART	GPT2	GPT2	0.9810	0.9810	0.9810	0.9793
					RoBERTa	0.9622	0.9624	0.9619	0.9603
					DistilBERT	**0.9885**	**0.9885**	**0.9885**	**0.9885**
				BART	BART	0.2500	0.2500	0.2500	0.0985
					RoBERTa	0.2485	0.2485	0.2485	0.0982
					DistilBERT	0.9563	0.9563	0.9563	0.9540
				BERT	BERT	0.9753	0.9771	0.9749	0.9746
					RoBERTa	0.9851	0.9851	0.9851	0.9845
					DistilBERT	0.9758	0.9759	0.9757	0.9748
			GAI-BERT	GPT2	GPT2	0.9866	0.9883	0.9845	0.9856
					RoBERTa	0.9675	0.9678	0.9671	0.9645
					DistilBERT	0.8275	0.8652	0.8071	0.7954
				BART	BART	0.9735	0.9756	0.9717	0.9729
					RoBERTa	0.2515	0.2515	0.2515	0.0992
					DistilBERT	0.9710	0.9707	0.9702	0.9684
				BERT	BERT	0.9833	0.9833	0.9833	0.9821
					RoBERTa	**0.9881**	**0.9882**	**0.9880**	**0.9873**
					DistilBERT	0.9868	0.9868	0.9864	0.9854

ficacy. Notably, it experiments with GPT-2, BART and BERT models as the primary stream within a dual model structure, coupled with various models in the secondary stream. The empirical results, however, revealed a marked under-performance when juxtaposed with setups where BART served as one of two streams. This recurrent pattern of subpar results, particularly evident in datasets synthesized by both GAI-GPT2 and GAI-BART, points sometimes to a potential inadequacy of the BART model within the context of current experimental framework. The consistently low accuracy scores associated with the BART model underscore its limitations for the tasks and datasets under consideration. This critical insight necessitates a more rigorous and nuanced approach in the selection and application of models for these specific types of computational tasks.

Additionally, it has been observed that dual models incorporating GPT2-X demonstrate a consistent and notable superiority in performance over their BART-X and BERT-X counterparts. This indicates a distinct advantage of GPT2 in the realm of text classification, especially concerning texts generated by both GPT2, BERT and BART, in contrast to the results achieved with BART and BERT models. Such findings raise critical considerations regarding the efficacy and applicability of these models in specific text classification scenarios.

Furthermore, it is imperative to address the influence of dataset characteristics on model performance. This research reveals that the dual model configuration GPT2-X

showed good performance for all generated datasets. This observation may suggest that the BART model possesses a superior capability in the context of data generation tasks. Such a differential impact underscores the importance of dataset selection and its consequential effect on the performance metrics of various model configurations.

Finally, the model associated with the second stream (which varied between RoBERTa, DistilBERT, and a duplicate of the first stream) significantly influenced the performance of the primary model in the first stream. Indeed, RoBERTa and DistilBERT slightly degraded the performance of the other models but remain robust when applied to GAI-GPT2 and GAI-BERT. However, when applied to GAI-BART, the performance degradation is more pronounced. This could be interpreted as poor generation by BART, whereas the generation quality by GPT-2 and BERT is much better than that generated by BART.

5.3.7 Single-stream vs. dual-stream models

To clearly demonstrate the effectiveness of the proposed dual-stream models compared to existing single-stream approaches, we conducted a detailed statistical analysis of the performance metrics (accuracy, precision, recall, and F1 score) across various datasets. Our findings show that dual-stream models consistently outperform single-stream models, as evidenced by significant improvements in all performance metrics. We performed paired t-tests which revealed statistically significant p-values (less to 0.05) for accuracy, precision, recall, and F1 score. Additionally, we calculated 95% confidence intervals for these metrics, demonstrating the reliability and robustness of our results. The dual-stream models not only achieved higher mean performance but also exhibited lower variability, suggesting more consistent results across different datasets. These findings highlight the significant potential of dual-stream architectures in improving the detection and classification of fake news.

6 Conclusion

The contributions of this research are threefold. The initial segment delved into surveying the influence of fake news on Internet users and their ability to discern it. Despite employing both Big five criteria and the SEM method, the survey revealed consistently low rates, indicating that information consumers struggle to detect generated fake content, scoring an average of 2.73 within the [0.6] interval. This underscores the elevated performance of fake news generation, rendering it challenging for Internet users to identify them in the majority of cases.

The second phase centered around generating a novel fake news dataset encompassing four types: fake, real, GAI-fake, and GAI-real. Leveraging various standard generators, performance evaluation highlighted BERT's unparalleled efficacy, showcasing impressive metrics with ROUGE (91.6%) and BERT-Score (91.8%).

Conclusively, the third segment demonstrated that our DuSTraMo model classifying the four ERAF-News classes significantly improved the detection performance ensuring an accuracy of 99.12%. Integrating our four-category classification model into existing fake news detection systems improves their accuracy and effectiveness while recognizing nuanced information to better combat misinformation.

While our four-category classification model offers many advantages, it is essential to: (1) control the complexity of the model when adding additional news categories, and (2) consider the ability of DuSTraMo to generalize effectively across languages using Transfer Learning and Federated Learning.

Such research could be considered a starting point towards the future of multi-classes fake news detection, with the possibility of utilizing additional models for comparative purposes.

Declarations

- Declaration of generative AI and AI-assisted technologies in the writing process : During the preparation of this work, sometimes the authors used ChatGPT in order to improve writing texts. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the content of the publication.
- Funding: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- Conflict of interest: On behalf of all authors, the corresponding author states that there is no conflict of interest.
- Ethics approval: Not applicable
- Consent to participate: Not applicable
- Consent for publication: Not applicable
- Availability of data and materials: A shared drive is available at the link https://rb.gy/kbt7lv.
- Code availability: Not applicable
- Methods: For the writing of the paper, chatGPT was sometimes used.
- Authors' contributions: In Table 8

Table 8: Authors' contributions

Authors	Supervision	Bibliographic search	Coding	Discussion results	Writing paper	Content paper
Hounaida Moalla		X	X	X	X	X
Hana Abid		X			X	X
Dorsaf Sallami		X	X	X	X	X
Esma Aïmeur	X					X
Bassem Ben Hamed	X					X

References

[1] Velichety, S., & Shrivastava, U. (2022). Quantifying the impacts of online fake news on the equity value of social media platforms–Evidence from Twitter. International Journal of Information Management, 64, 102474. https://doi.org/10.1016/j.ijinfomgt.2022.102474

[2] Gupta, M., Dennehy, D., Parra, C. M., Mäntymäki, M., & Dwivedi, Y. K. (2023). Fake news believability: The effects of political beliefs and espoused cultural values. Information & Management, 60(2), 103745. https://doi.org/10.1016/j.im.2022.103745

[3] Choi, J., & Lee, J. K. (2022). Confusing effects of fake news on clarity of political information in the social media environment. Journalism Practice, 16(10), 2147-2165. https://doi.org/10.1080/17512786.2021.1903971

[4] Loos, E., & Ivan, L. (2022). Fighting Fake News: A Generational Approach (p. 172). MDPI-Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/soc12020057

[5] Baptista, J.P.; Gradim, A. (2022). A Working Definition of Fake News. Encyclopedia 2022, 2, 632-645. https://doi.org/10.3390/encyclopedia2010043

[6] Leeder, C. (2019). How college students evaluate and share "fake news" stories. Library & Information Science Research, 41(3), 100967. https://doi.org/10.1016/j.lisr.2019.100967

[7] Abu Arqoub, O., Abdulateef Elega, A., Efe Özad, B., Dwikat, H., & Adedamola Oloyede, F. (2022). Mapping the scholarship of fake news research: A systematic review. Journalism Practice, 16(1), 56-86. https://doi.org/10.1080/17512786.2020.1805791

[8] Sallami, D. (2022). Personalized fake news aware recommendation system. https://doi.org/1866/27492

[9] Amri, S., Sallami, D., & Aïmeur, E. (2021). Exmulf: an explainable multimodal content-based fake news detection system. In International Symposium on Foundations and Practice of Security (pp. 177-187). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-08147-7_12

[10] Sallami, D., Ben Salem, R., & Aïmeur, E. (2023). Trust-based Recommender System for Fake News Mitigation. In Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization (pp. 104-109). https://doi.org/10.1145/3563359.3597395

[11] Kondamudi, M. R., Sahoo, S. R., Chouhan, L., & Yadav, N. (2023). A comprehensive survey of fake news in social networks: Attributes, features, and detection approaches. Journal of King Saud University-Computer and Information Sciences, 35(6), 101571. https://doi.org/10.1016/j.jksuci.2023.101571

[12] Gao, Y., González, V. A., & Yiu, T. W. (2020). Exploring the relationship between construction workers' personality traits and safety behavior. Journal of construction engineering and management, 146(3), 04019111. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001763

[13] Qian, K., & Yahara, T. (2020). Mentality and behavior in COVID-19 emergency status in Japan: Influence of personality, morality and ideology. PloS one, 15(7), e0235883. https://doi.org/10.1371/journal.pone.0235883

[14] Al Ayub Ahmed, A., Aljabouh, A., Donepudi, P. K., & Suh Choi, M. (2021). Detecting Fake News Using Machine Learning: A Systematic Literature Review. arXiv-2102. https://doi.org/10.48550/arXiv.2102.04458

[15] Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. Machine Learning with Applications, 4, 100032. https://doi.org/10.1016/j.mlwa.2021.100032

[16] Faustini, P. H. A., & Covoes, T. F. (2020). Fake news detection in multiple platforms and languages. Expert Systems with Applications, 158, 113503.

https://doi.org/10.1016/j.eswa.2020.113503

[17] Farokhian, M., Rafe, V., & Veisi, H. (2023). Fake news detection using dual BERT deep neural networks. Multimedia Tools and Applications, 1-18. https://doi.org/10.1007/s11042-023-17115-w

[18] Ibrishimova, M. D., & Li, K. F. (2020). A machine learning approach to fake news detection using knowledge verification and natural language processing. In Advances in Intelligent Networking and Collaborative Systems: The 11th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2019) (pp. 223-234). Springer International Publishing. https://doi.org/10.1007/978-3-030-29035-1_22

[19] Kula, S., Choraś, M., & Kozik, R. (2021). Application of the BERT-based architecture in fake news detection. In 13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020) 12 (pp. 239-249). Springer International Publishing. https://doi.org/10.1007/978-3-030-57805-3_23

[20] Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. Advances in neural information processing systems, 32. https://shorturl.at/mBHL7

[21] Schuster, T., Schuster, R., Shah, D. J., & Barzilay, R. (2020). The limitations of stylometry for detecting machine-generated fake news. Computational Linguistics, 46(2), 499-510. https://doi.org/10.1162/coli_a_00380

[22] Fröhling, L., & Zubiaga, A. (2021). Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. PeerJ Computer Science, 7, e443. https://doi.org/10.7717/peerj-cs.443

[23] Karimi, H., Roy, P., Saba-Sadiya, S., & Tang, J. (2018, August). Multi-source multi-class fake news detection. In Proceedings of the 27th international conference on computational linguistics (pp. 1546-1557).

[24] Pennycook, G., & Rand, D. G. (2021). The psychology of fake news. Trends in cognitive sciences, 25(5), 388-402. https://doi.org/10.1016/j.tics.2021.02.007

[25] Ansar, W., & Goswami, S. (2021). Combating the menace: A survey on characterization and detection of fake news from a data science perspective. International Journal of Information Management Data Insights, 1(2), 100052. https://doi.org/10.1016/j.jjimei.2021.100052

[26] Osmundsen, M., Bor, A., Vahlstrup, P. B., Bechmann, A., & Petersen, M. B. (2021). Partisan polarization is the primary psychological motivation behind political fake news sharing on Twitter. American Political Science Review, 115(3), 999-1015. https://doi.org/10.1017/S0003055421000290

[27] Igwebuike, E. E., & Chimuanya, L. (2021). Legitimating falsehood in social media: A discourse analysis of political fake news. Discourse & Communication, 15(1), 42-58. https://doi.org/10.1177/1750481320961659

[28] Li, Q., Hu, Q., Lu, Y., Yang, Y., & Cheng, J. (2020). Multi-level word features based on CNN for fake news detection in cultural communication. Personal and Ubiquitous Computing, 24, 259-272. https://doi.org/10.1007/s00779-019-01289-y

[29] Dabbous, A., Aoun Barakat, K., & de Quero Navarro, B. (2022). Fake news detection and social media trust: a cross-cultural perspective. Behaviour & Information Technology, 41(14), 2953-2972. https://doi.org/10.1080/0144929X.2021.1963475

[30] Petratos, P. N. (2021). Misinformation, disinformation, and fake news: Cyber risks to business. Business Horizons, 64(6), 763-774. https://doi.org/10.1016/j.bushor.2021.07.012

[31] Fong, B. (2021). Analysing the behavioural finance impact of'fake news' phenomena on financial markets: a representative agent model and empirical validation. Financial Innovation, 7(1), 1-30. https://doi.org/10.1186/s40854-021-00271-z

[32] Obadă, D. R., & Dabija, D. C. (2022). Can Fake News About Companies Lead to an Increased Social Media Usage? An Empirical Investigation. https://philpapers.org/rec/OBACFN

[33] Radwan, E., Radwan, A., & Radwan, W. (2020). The role of social media in spreading panic among primary and secondary school students during the COVID-19 pandemic: An online questionnaire study from the Gaza Strip, Palestine. Heliyon, 6(12). https://doi.org/10.1016/j.heliyon.2020.e05807

[34] Weiss, A. P., Alwan, A., Garcia, E. P., & Garcia, J. (2020). Surveying fake news: Assessing university faculty's fragmented definition of fake news and its impact on teaching critical thinking. International Journal for Educational Integrity, 16, 1-30. https://doi.org/10.1007/s40979-019-0049-x

[35] Baptista, J. P., Correia, E., Gradim, A., & Piñeiro-Naval, V. (2021). The influence of political ideology on fake news belief: The Portuguese case. Publications, 9(2), 23. https://doi.org/10.3390/publications9020023

[36] Zanatta, E. T., Wanderley, G. P. D. M., Branco, I. K., Pereira, D., Kato, L. H., & Maluf, E. M. C. P. (2021). Fake news: The impact of the internet on population health. Revista da Associação Médica Brasileira, 67, 926-930. https://doi.org/10.1590/1806-9282.20201151

[37] Pérez-Escoda, A., Pedrero-Esteban, L. M., Rubio-Romero, J., & Jiménez-Narros, C. (2021). Fake news reaching young people on social networks: Distrust challenging media literacy. Publications, 9(2), 24. https://doi.org/10.3390/publications9020024

[38] Sampat, B., & Raj, S. (2022). Fake or real news? Understanding the gratifications and personality traits of individuals sharing fake news on social media platforms. Aslib Journal of Information Management, 74(5), 840-876. https://doi.org/10.1108/AJIM-08-2021-0232

[39] Torabi, M., & Sotudeh, H. (2022). The Role of Risk Perception and Ability to Detect Fake News in Acceptance of COVID-19 Vaccine among Students of Shiraz University, Iran. Health Information Management, 18(6), 265-271. https://doi.org/10.22122/him.v18i1.4440

[40] Chuai, Y., & Zhao, J. (2022). Anger can make fake news viral online. Frontiers in Physics, 10, 970174. https://doi.org/10.3389/fphy.2022.970174

[41] Tan, W. K., & Hsu, C. Y. (2023). The application of emotions, sharing motivations, and psychological distance in examining the intention to share COVID-19-related fake news. Online Information Review, 47(1), 59-80. https://doi.org/10.1108/OIR-08-2021-0448

[42] Guo, Z., Schlichtkrull, M., & Vlachos, A. (2022). A survey on automated fact-checking. Transactions of the Association for Computational Linguistics, 10, 178-206. https://doi.org/10.1162/tacl_a_00454

[43] Ghadiri, Z., Ranjbar, M., Ghanbarnejad, F., & Raeisi, S. (2022). Automated Fake News Detection using cross-checking with reliable sources. arXiv:2201.00083. https://doi.org/10.48550/arXiv.2201.00083

[44] Himdi, H., Weir, G., Assiri, F., & Al-Barhamtoshy, H. (2022). Arabic fake news detection based on textual analysis. Arabian Journal for Science and Engineering, 47(8), 10453-10469. https://doi.org/10.1007/s13369-021-06449-y

[45] Mazari, A. C., & Djeffal, A. (2022). Sentiment analysis of Algerian dialect using machine learning and deep learning with Word2vec. Informatica, 46(6). https://doi.org/10.31449/inf.v46i6.3340

[46] Hu, L., Wei, S., Zhao, Z., & Wu, B. (2022). Deep learning for fake news detection: A comprehensive survey. AI Open. https://doi.org/10.1016/j.aiopen.2022.09.001

[47] Sallami, D., Gueddiche, A., & Aïmeur, E. (2023). From Hype to Reality: Revealing the Accuracy and Robustness of Transformer-Based Models for Fake News Detection. https://ceur-ws.org/Vol-3593/paper2.pdf

[48] de Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V., & Mattos, D. M. (2021). Identifying fake news on social networks based on natural language processing: trends and challenges. Information, 12(1), 38. https://doi.org/10.3390/info12010038

[49] Shu, K., Wang, S., Lee, D., & Liu, H. (2020). Mining disinformation and fake news: Concepts, methods, and recent advancements. Disinformation, misinformation, and fake news in social media: Emerging research challenges and opportunities, 1-19. https://doi.org/10.1007/978-3-030-42699-6_1

[50] Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management, 57(2), 102025. https://doi.org/10.1016/j.ipm.2019.03.004

[51] Aïmeur, E., Amri, S., & Brassard, G. (2023). Fake news, disinformation and misinformation in social media: a review. Social Network Analysis and Mining, 13(1), 30. https://doi.org/10.1007/s13278-023-01028-5

[52] Natarajan, R., Mehbodniya, A., Rane, K. P., Jindal, S., Hasan, M. F., Vives, L., & Bhatt, A. (2022). Intelligent gravitational search random forest algorithm for fake news detection. International Journal of Modern Physics C, 33(06), 2250084. https://doi.org/10.1142/S012918312250084X

[53] Hussain, M. G., Hasan, M. R., Rahman, M., Protim, J., & Al Hasan, S. (2020). Detection of bangla fake news using mnb and svm classifier. In 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE) (pp. 81-85). IEEE. https://doi.org/10.1109/iCCECE49321.2020.9231167

[54] Jain, P., Sharma, S., & Aggarwal, P. K. (2022). Classifying fake news detection using SVM, Naive Bayes and LSTM. In 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 460-464). IEEE. https://doi.org/10.1109/Confluence52989.2022.9734129

[55] Tourille, J., Sow, B., & Popescu, A. (2022, June). Automatic Detection of Bot-generated Tweets. In Proceedings of the 1st International Workshop on Multimedia AI against Disinformation (pp. 44-51). https://doi.org/10.1145/3512732.3533584

[56] Aladeen, H. (2023). Breaking News: Machine Learning Helps to Spot Fake News Before it Spreads. IConFN'23, June 03–05, 2023, Wadiya.

[57] Biradar, S., Saumya, S., & Chauhan, A. (2023). Combating the infodemic: COVID-19 induced fake news recognition in social media networks. Complex & Intelligent Systems, 9(3), 2879-2891. https://doi.org/10.1007/s40747-022-00672-2

[58] Hutama, L. B., & Suhartono, D. (2022). Indonesian hoax news classification with multilingual transformer model and BERTopic. Informatica, 46(8). https://doi.org/10.31449/inf.v46i8.4336

[59] Adoma, A. F., Henry, N. M., & Chen, W. (2020). Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP) (pp. 117-121). IEEE. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317379

[60] Wagh, V., Khandve, S., Joshi, I., Wani, A., Kale, G., & Joshi, R. (2021). Comparative study of long document classification. In TENCON 2021-2021 IEEE Region 10 Conference (TENCON) (pp. 732-737). IEEE. https://doi.org/10.1109/TENCON54134.2021.9707465

[61] Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. International Journal of Information Management Data Insights, 1(1), 100007. https://doi.org/10.1016/j.jjimei.2020.100007

[62] Yang, Z., Ma, J., Chen, H., Lin, H., Luo, Z., & Chang, Y. (2022). A Coarse-to-fine Cascaded Evidence-Distillation Neural Network for Explainable Fake News Detection. arXiv:2209.14642. https://doi.org/10.48550/arXiv.2209.14642

[63] Pan, Y., Pan, L., Chen, W., Nakov, P., Kan, M. Y., & Wang, W. Y. (2023). On the Risk of Misinformation Pollution with Large Language Models. arXiv:2305.13661. https://doi.org/10.48550/arXiv.2305.13661

[64] Sitaula, N., Mohan, C. K., Grygiel, J., Zhou, X., & Zafarani, R. (2020). Credibility-based fake news detection. Disinformation, misinformation, and fake news in social media: Emerging research challenges and Opportunities, 163-182. https://doi.org/10.1007/978-3-030-42699-6_9

[65] Barberá, P., Boydstun, A. E., Linn, S., McMahon, R., & Nagler, J. (2021). Automated text classification of news articles: A practical guide. Political Analysis, 29(1), 19-42. https://doi.org/10.1017/pan.2020.8

[66] Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv:2005.14165. https://shorturl.at/BJQX4

[67] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv:2303.08774. https://doi.org/10.48550/arXiv.2303.08774

[68] Yanagi, Y., Orihara, R., Sei, Y., Tahara, Y., & Ohsuga, A. (2020, July). Fake news detection with generated comments for news articles. In 2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES) (pp. 85-90). IEEE. https://doi.org/10.1109/INES49302.2020.9147195

[69] Kreps, S., McCain, R. M., & Brundage, M. (2022). All the news that's fit to fabricate: AI-generated text as a tool of media misinformation. Journal of experimental political science, 9(1), 104-117. https://doi.org/10.1017/XPS.2020.37

[70] Xu, D., Fan, S., & Kankanhalli, M. (2023). Combating misinformation in the era of generative AI models. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 9291-9298). https://doi.org/10.1145/3581783.3612704

[71] Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., ... & Wang, J. (2019). Release strategies and the social impacts of language models. arXiv:1908.09203. https://doi.org/10.48550/arXiv.1908.09203

[72] Kumarage, T., Garland, J., Bhattacharjee, A., Trapeznikov, K., Ruston, S., & Liu, H. (2023). Stylometric detection of ai-generated text in twitter timelines. arXiv:2303.03697. https://doi.org/10.48550/arXiv.2303.03697

[73] Najee-Ullah, A., Landeros, L., Balytskyi, Y., & Chang, S. Y. (2021). Towards detection of AI-generated texts and misinformation. In International Workshop on Socio-Technical Aspects in Security (pp. 194-205). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-10183-0_10

[74] Chang, S. Y. (2022). Towards detection of AI-generated texts and misinformation. In Socio-Technical Aspects in Security: 11th International Workshop, STAST 2021, Virtual Event (p. 194). Springer Nature. https://doi.org/10.1007/978-3-031-10183-0

[75] Gillioz, A., Casas, J., Mugellini, E., & Abou Khaled, O. (2020, September). Overview of the Transformer-based Models for NLP Tasks. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS) (pp. 179-183). IEEE. https://doi.org/10.15439/2020F20

[76] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9. https://shorturl.at/cCL23

[77] Yu, Y., Zhan, F., Wu, R., Pan, J., Cui, K., Lu, S., ... & Miao, C. (2021). Diverse image inpainting with bidirectional and autoregressive transformers. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 69-78). https://doi.org/10.1145/3474085.3475436

[78] Kenton, J. D. M. W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of naacL-HLT (Vol. 1, p. 2). https://shorturl.at/gyGMS

[79] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108

[80] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692. https://doi.org/10.48550/arXiv.1907.11692

[81] Papapicco, C., Lamanna, I., & D'Errico, F. (2022). Adolescents' vulnerability to fake news and to racial hoaxes: a qualitative analysis on italian sample. Multimodal Technologies and Interaction, 6(3), 20. https://doi.org/10.3390/mti6030020

[82] Arin, K. P., Mazrekaj, D., & Thum, M. (2023). Ability of detecting and willingness to share fake news. Scientific Reports, 13(1), 7298. https://doi.org/10.1038/s41598-023-34402-6

[83] Peng, Y., Pei, C., Zheng, Y., Wang, J., Zhang, K., Zheng, Z., & Zhu, P. (2020). A cross-sectional survey of knowledge, attitude and practice associated with COVID-19 among undergraduate students in China. BMC public health, 20(1), 1-8. https://doi.org/10.1186/s12889-020-09392-z

[84] Mehta, Y., Majumder, N., Gelbukh, A., & Cambria, E. (2020). Recent trends in deep learning based personality detection. Artificial Intelligence Review, 53, 2313-2339. https://doi.org/10.1007/s10462-019-09770-z

[85] Shi, D., & Maydeu-Olivares, A. (2020). The effect of estimation methods on SEM fit indices. Educational and psychological measurement, 80(3), 421-445. https://doi.org/10.1177/0013164419885

[86] Schreiber, J. B. (2022). Key processes and popular analyses in the SEM family of techniques. Contemporary Research Methods in Pharmacy and Health Services, 601-616. https://doi.org/10.1016/B978-0-323-91888-6.00023-5

[87] Zyphur, M. J., Bonner, C. V., & Tay, L. (2023). Structural equation modeling in organizational research: The state of our science and some proposals for its future. Annual Review of Organizational Psychology and Organizational Behavior, 10, 495-517. https://doi.org/10.1146/annurev-orgpsych-041621-031401

[88] Petropoulos, P., & Petropoulos, V. (2024). RoBERTa and Bi-LSTM for Human vs AI Generated Text Detection. Working Notes of CLEF.

[89] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:1910.13461. https://doi.org/10.48550/arXiv.1910.13461

[90] Kang, H. L., Na, S. D., & Kim, M. N. (2021). A method for enhancing speech and warning signals based on parallel convolutional neural networks in a noisy environment. Technology and Health Care, 29(S1), 141-152. https://doi.org/10.3233/THC-218015

[91] Cai, S., Han, D., Yin, X., Li, D., & Chang, C. C. (2022). A hybrid parallel deep learning model for efficient intrusion detection based on metric learning. Connection Science, 34(1), 551-577. https://doi.org/10.1080/09540091.2021.2024509

[92] Song, Y. F., Zhang, Z., Shan, C., & Wang, L. (2020, October). Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In proceedings of the 28th ACM international conference on multimedia (pp. 1625-1633). https://doi.org/10.1145/3394171.3413802

[93] Singh, A., Sengupta, S., & Lakshminarayanan, V. (2020). Explainable deep learning models in medical image analysis. Journal of imaging, 6(6), 52. https://doi.org/10.3390/jimaging6060052

[94] Ras, G., Xie, N., Van Gerven, M., & Doran, D. (2022). Explainable deep learning: A field guide for the uninitiated. Journal of Artificial Intelligence Research, 73, 329-396. https://doi.org/10.1613/jair.1.13200

[95] Singh, N., Kaliyar, R. K., Vivekanand, T., Uthkarsh, K., Mishra, V., & Goswami, A. (2022). B-LIAR: A novel model for handling Multiclass Fake News data utilizing a Transformer Encoder Stack-based architecture. In 2022 1st International Conference on Informatics (ICI) (pp. 31-35). IEEE. https://doi.org/10.1109/ICI53355.2022.9786925

[96] Nield, T. (2022). Essential Math for Data Science. " O'Reilly Media, Inc.". https://shorturl.at/ekvxH

[97] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. arXiv:1904.09675. https://doi.org/10.48550/arXiv.1904.09675

[98] Sellam, T., Das, D., & Parikh, A. P. (2020). BLEURT: Learning robust metrics for text generation. arXiv:2004.04696. https://doi.org/10.48550/arXiv.2004.04696

[99] Zhou, M., Li, Z., Tan, B., Zeng, G., Yang, W., He, X., ... & Xie, P. (2021). On the generation of medical dialogs for COVID-19. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). https://par.nsf.gov/servlets/purl/10345461

[100] Yuan, W., Neubig, G., & Liu, P. (2021). Bartscore: Evaluating generated text as text generation. Advances in Neural Information Processing Systems, 34, 27263-27277. https://shorturl.at/adqX0

[101] He, T., Zhang, J., Wang, T., Kumar, S., Cho, K., Glass, J., & Tsvetkov, Y. (2022). On the blind spots of model-based evaluation metrics for text generation. arXiv:2212.10020. https://doi.org/10.48550/arXiv.2212.10020

[102] Hunt, J. (2019). A beginners guide to Python 3 programming. Springer. https://doi.org/10.1007/978-3-030-20290-39

[103] Petrelli, M. (2021). Introduction to Python in Earth Science Data Analysis: From Descriptive Statistics to Machine Learning. Springer Nature. https://doi.org/10.1007/978-3-030-78055-5

[104] Bhattarai, B., Granmo, O. C., & Jiao, L. (2021). Explainable tsetlin machine framework for fake news detection with credibility score assessment. arXiv preprint arXiv:2105.09114. https://doi.org/10.48550/arXiv.2105.09114

[105] Guo, M., Liu, L., Guo, M., Liu, S., & Xu, Z. (2023). Accurate Generated Text Detection Based on Deep Layer-wise Relevance Propagation. In 2023 IEEE 8th International Conference on Big Data Analytics (ICBDA) (pp. 215-223). IEEE. https://doi.org/10.1109/ICBDA57405.2023.10104941

A Novel Music Recommendation System Using Filtering Techniques

Srishti Vashishtha[1], Deepika Varshney[2], Eva Sarin[3], Simran Kaur[4]
[1,3,4]Computer science and engineering dept., Bharati Vidyapeeth's College of Engineering, GGSIPU New Delhi, India
[2]Computer science and engineering dept., Jaypee Institute of information Technology, Noida, India
E-mail: deepikavarshney06@gmail.com

Keywords: music recommendation system, content-based filtering, collaborative filtering, TFIDF, KNN, cosine similarity

Received: November 17, 2023

With the growth of the World Wide Web, a large amount of music data is available on the Internet. A large number of people listen to music online rather than downloading and listening offline. But only some sites provide personalized and accurately recommended songs while they listen to an auto-playing playlist. Hence the need for recommendation systems arises. Two approaches can be applied to the recommendation system: content based filtering and collaborative filtering. While in content-based filtering approach, analysis on the songs' content which has been preferred by the user in history is done and the songs with relative similarity are recommended. While latter suggests songs that certain users of similar listening pattern have preferred. But collaborative filtering-based recommendation systems not only requires time to attain stability but also might recommend unsuitable music because it is not personalized to each user's preference. Also, the latter requires the songs to be listened by a few users already to recommend it any other user. Hence for overall analysis, answers to few questions need to be implemented in recommendation systems: the very first one is how the properties should be analyzed, next one is how the analysis should be done and last is how the songs related to the user's preference should be chosen. So, for suggesting the better system which solves these three questions as well as provides better and personalized recommendations, In this paper we build a collaborative filtering as well as content based filtering recommendation system, where various factors determine the essence of the songs, i.e. liveliness, keys used, loudness, pitch, valence, etc., are analyzed for comparison. From the experimental analysis in has been identified that content-based filtering technique performed best on KNN machine learning classifier with accuracy of 85%.

Povzetek: Članek predstavi nov sistem za priporočanje glasbe, ki uporablja tehnike filtriranja na osnovi vsebine in sodelovalno filtriranje. Z uporabo metod, kot sta TF-IDF in KNN, raziskuje izboljšanje personaliziranih priporočil za uporabnike.

1 Introduction

Music is an important vehicle for communicating to other people something relevant about our personality, history, etc. [9]. Music files are easily available on websites for free in this era. To this date, every single online music site holds over 2,000,000 songs [1]. More than 1,000 songs are available on storages of various personal computers (PCs) or hard disks equipped in MP3 players [5]. But only a few provide relevant music recommendations from this huge amount of music data available. Hence it becomes necessary to generate personalized recommendations using music recommendation systems [2]. For listeners, when choosing a new song to listen to, the song's contained affective content plays a huge role in its selection which should be as per their preference and music taste. However, when users have to pick a song from their current playlist, they often tend to rely on their current emotions to select the song they feel they want to listen to the most. Thus, a need for a music recommendation system prevails that not only understands the current feelings of the user but also understands what user might want to listen in similar moods. [8]. Music recommendation systems can be based on a lot of factors, on a user's facial expressions [7], past music listening behavior [11], current emotions [15], or other user's preferences based on sentiments [44] as well. The job of such system is filtering out every piece of content that is available and provides with only that part of information which provides interested content. [6]. The two approaches to such recommendation systems: 1. Content-based filtering and collaborative filtering. Collaborative filtering is based on the idea that similar users tend to behave in similar manner in similar situations [6]. Hence, it considers the actions, past behavior, user history, and preferences of other users for similar songs to the user of interest. While the idea behind the content-based filtering method offers the user of interest similar subjects as listened to by him previously. The difference between the two approaches is that in the latter the similarities between two songs is not evaluated

on user actions, which is the case in the former, but on the features of the object, i.e. song itself.

Music recommendation systems can be called information retrieval tools, as well. [25] It means that there is a certain information demand in the audience which needs to be dealt with. While making a recommendation system, dealing with this demand becomes difficult since most of the time, the user in interest doesn't always know what type of music, in particular, he or she wants to listen to. In such cases, both the approaches mentioned above provide quality recommendations. But music is not objective and accepted universally [26]. Music conveys emotion and at the same moment plays a major role in changing the mood of the listener. Therefore, one cannot always rely on only user history for generating recommendations, but consider a comparison with different songs for similarities as well.

A large number of people listen to music online rather than downloading and listening offline. But only some sites provide personalized and accurately recommended songs while they listen to an auto-playing playlist. The purpose of this research work is to provide music recommendations to listeners and answer to few questions. The questions are: how the properties should be analyzed, next one is how the analysis should be done and last is how the songs related to the user's preference should be chosen.

Hence, this paper surveys both collaborative, as well as content-based filtering, approaches for comparison, which generates better and quality recommendations in terms of variety, genre, themes as well as emotions. For content-based filtering, various mathematical tools like TF IDF and cosine similarity will help in the generation of recommendations, while in the case of collaborative filtering, recommendation systems based on machine learning algorithms like KNN will be experimented on for generating recommendations.

The paper is organized in the mentioned sequence: Section 1 contains introduction, Section 2 has the literature review of recent works, Section 3 describes the proposed architecture of recommendation systems, the experimental setups and dataset description, Section 4 discusses the results, Section 5 is about the conclusion drawn along with future work followed by references.

2 Related work

The authors in [1] proposed a mechanism for providing tailored services. Authors employed STFT (Shortest Time Fourier Form) to examine properties of music. They presented a dynamic K-means clustering technique to examine users' preferences, which clustered the songs in the music list dynamically changing number of clusters. They created a music database with 100 songs from ballad, rock, jazz, and classical genres. Each piece's feature was divided by the average, which set the maximum radius (Rmax) to 0.5 and the cell size to 0.2. 80% of recommended pieces were near to the users' desire. Rock and classical music are the most frequently recommended genres. The user's preferred genres account for 74% of the total (34 percent rock + 40% classical).

The Music Suggestion System (MRS) was created to provide a personalized music recommendation service [2]. MIDI format musical instruments were first tested. The feature track of each polyphonic musical element was first determined, followed by six elements in the track. Use music object parameters that include tone, duration, and sound. Of the 100 MIDI files, the results showed an accuracy of 83%. The K-means algorithm has been used to classify the same database based on six recovered elements.

The DJ-Running initiative is a research endeavor aimed at tracking both the mental and physical exertion of athletes during training sessions to comprehend their emotional state. Its goal is to promptly curate the most fitting music selections to enhance motivation and performance. This article [3] outlines a renowned music support system devised within this project, which tailors the next song choice based on factors like the user's location, mood, and activity type. Leveraging Spotify's search technology, the system interprets user cues and transforms them into recommendation algorithms. The underlying database encompasses three key services: user data, geographic data, and music distribution, all constructed using the Spring framework. In a separate study by the authors of [4], they introduced a unified music recognition system coupled with automated genre classification and emotion-based categorization. This system employs frequency coefficients adjusted for logarithmic scale modulation, a novel and expedited characteristic. All relevant features are derived from the MP3 cutting tool, effectively halving the extraction time. The primary challenge proposed is to amalgamate and enhance the efficacy of music classification and classification theory through the utilization of the AdaBoost algorithm. The study utilized a dataset comprising 1000 songs (for uniform classification) and 800 songs (for emotional data). Results indicate a notable improvement in classification accuracy for five genres (classical, pop, hip-hop, soul, and punk), with accuracy climbing from 86.8% to 92.2% when compared to traditional methods. Accuracy of four-emotion classification (sad, calm, pleasant, and excited) was increased from 86.0% to 90.5%.

A content-based recommender system was developed by researchers in [5]. The system employs two strategies: 1. Acoustic feature analysis and 2. Application of deep learning and computer vision approaches to improve the recommender system's outcomes. The tasks were divided in 2 subtasks: A. In order to create an acceptable vector space representation of music composition of particular dimensions; B. In order to evaluate vector representations of songs. Distance metrics like Euclidean, Manhattan and Cosine Distances were used to determine the similarities between vector representations of the acoustic characteristics. The similarity between vector representations of auditory properties was determined using distances. Random recommendation, genre-specific random recommendations, auditory characteristics analysis, and artificial neural network all had precision of

0.006, 0.066, 0.112, and 0.148, respectively. Apparently, the quality of the direct integration of the vector expression type was high when using the neural network. The author's hypotheses that the vector representation created by ANN included more content about music were also validated by the results.

S Metilda Florence and M Uma in [6] presented users with suggestions that matched their interests. The authors hoped that by creating a recommendation system, they would be able to help a user decide which music to listen to in order to reduce stress levels. Emotion extraction module, Audio extraction module, and Emotion-Audio Integration Module were most commonly used methods. The HELEN dataset, which had over 2000 photos, was utilized to train the classifier which was used to detect facial landmarks from the user's face. The CK extensive dataset was utilized to determine user's expressed emotion. When this is detected, the music player chooses a song that best suits the user's mood. The system's accuracy was 80%. Disadvantage of this technology is that it must be tested in various lighting conditions to evaluate its robustness. If the classifier intends to determine the emotions of the user, the image quality must be at least 320p. The authors of [7] created a music recommendation system that tracks emotions of the user and offers songs by presenting a list of songs that is arranged appropriately based on the user's current emotion. The system ranks songs based on 2 criteria: relevance to the preferences of user; and influence on mentality of user's feelings. Sensors record the user's bio-signal data, which is subsequently used as input for the emotion detection process. Using RF-ECG sensor, the system was taught to determine emotional state of user based on temperature of skin and heartbeat rate. The algorithm extracted the user's emotion with an estimated accuracy of 64.5% according to results. 6.66% of songs were disliked, 8.57% songs were categorized as "bad influence" by users from the list of songs recommended to them. 3.6/5 is the average scale for EmuPlayer's work's satisfation. EmuPlayer fails to give accurate results and doesn't play the highest scored songs. In [8], researchers devised a recommendation mechanism leveraging Friend of a Friend (FOAF) and Rich Site Summary (RSS) vocabularies to suggest music to users based on various aspects of their musical taste, including psychological factors such as personality, demographic preferences, socio-economic status, and explicit musical inclinations. Web content was syndicated using XML format, with user preferences outlined through FOAF documents. User profiling, derived from the user's FOAF description, context-based information gleaned from music-related RSS feeds, and content-based descriptions extracted from the audio itself, facilitated music discovery. The system operated by: 1. Extracting interests from the user's FOAF profile; 2. Identifying artists and bands; 3. Selecting related artists based on those encountered in the user's FOAF profile; and 4. Ranking the results by relevance. The system achieved a 71.2 percent accuracy rate but fell short in accurately discerning the user's true emotions and musical preferences.

In [9], Yuri S. and Takayuki I. unveiled MusiCube, a music recommendation system that displayed a selection of songs as coloured icons in a 2-D cubic environment and provided a user interface for selecting suggested tunes intuitively. The authors demonstrated how MusiCube effectively depicted clouds of icons corresponding to sets of users' favourite tunes in a 2D cubic space. MusiCube initially calculates the feature values of tunes, then initialises the system, recommends numerous tunes by changing the colors of icons, receives user feedback, changes the colours of listening tunes' icons, and continues evolutionary computing to the next generation. The results are divided into three groups: positive (red), negative (orange), and not yet determined (yellow).

The authors' suggested system in [10] aimed to recommend music that the user like, are new to their ears, and fit the user's listening behaviour. For song recommendations, five factors were considered: genre, year, favour, freshness, and time pattern. The likelihood of playing a song at that time is computed using a Gaussian Mixture Model, which represents time pattern of listening. To forecast the genre, a web wrapper was created to collect genre information from AllMusic.com (a website that uses information to determine song genres). The ARIMA algorithm was used to forecast the next probable year and calculate the likelihood of a recording year. The Forgetting Curve was used to determine how fresh a song was to a user. The system was effective and accurate. (T-RECSYS) created an accurate recommendation system with real-time prediction by combining content-based and collaborative filtering as input to a deep learning classification model. The authors of [11] used data from Spotify Recsys Challenge to test their method. For content-based filtering, T-RECSYS considers six areas of metadata: genre, artist type, artist era, mood, tempo, and release year. T-RECSYS calculated the Sorenson index for each pair for collaborative filtering. Content-based and collaborative filtering information were combined to create input vector. The model was built, trained, and deployed using Google's Tensorflow and the Keras Python library. T-RECSYS can be used with Amazon, iTunes, Netflix, and other services. Real-time updates and many variable inputs were missing from the model.

To improve the Content Based technique, authors in [12] presented the TICI (Transaction-Interest-Count-Interest) method. They used two parameters: Count-Interest and Transaction-Interest, to allow users to choose which weight from the options offered they want to highlight. The CB approach could discover the most recently popular music group based on the user's access history; however, the result was unbalanced. Authors' proposed formula highlighted the importance of time as well as the number of musicians in a group. They place a high value on the passage of time. The TICI approach took into account the number of members in a music group and when they first appeared, allowing it to determine the group's weight rank more precisely than the CB method. This factor and system proved that TICI is a

more accurate and efficient method than the CB method. Authors aimed to offer a better formula (COL-Collaborative technique) for successful music data grouping than the CB method [13]. The COL technique would result in the users being grouped together because the supports of the groups with varied densities are the same. As a result, the TICI approach was proposed to improve the CB method, while the DI (Density-Interest) method was proposed to improve the COL method. The DI approach calculated music group support and took into account the distributions of the music group's appearances. In terms of weight difference, the results confirmed that the TICI approach could outperform the CB method. In terms of Hamming distance, the DI technique may outperform the COL method because DI considers the density of the appearance of the music group, allowing DI to distinguish users with various access behaviors more clearly than the COL method.

In comparison to the brute-force method, the researchers presented a method of measuring the acoustic distance between pre-classified music files with same sort of emotion in [14]. This method considerably sped up the search process and increased precision. The AdaBoost method was used to classify the music signature derived from the entire music database into predetermined music emotion classes. A combination of numerous elements is the main feature of the music database used. Sad, peaceful, pleasant, and exciting were the four music emotions evaluated. The great precision of music classification led to a higher recall rate and search speed than the previous brute-force method.

The authors add time scheduling to the music playlist to the already existing approaches, and incorporated decision tree categorization technology to help people find music that suits them better. A music recommendation system with Hybrid time scheduling was built in a web environment using decision-tree classification learning as its fundamental architecture. In the initial step, the system collects personal data such as gender, age, occupation, and favoured music genres. The feedback from users was then captured, including the marked time, content elements, and evaluation information. In the second step, groups of people with similar tastes were joined together via collaborative filtering. Player platform was developed using C#, system interface using Flash to build platform of MuPa recommendation system. The overall precision of the system is 78.33%. The diverse set of approaches for music recommendation system is shown in Table 1.

Wang et al. (2023) proposed a Multi-view Enhanced Graph Attention Network (named MEGAN) for session-based music recommendation. MEGAN can learn informative representations (embeddings) of music pieces and users from heterogeneous information based on graph neural network and attention mechanism. MEGAN achieves better performance than baselines, including several state-of-the-art recommendation methods [45]. Liu et al. designed a novel emotion-based personalized music recommendation framework to meet users' emotional needs and help improve their mental status. In this framework, authors designed a LSTM-based model to select the most suitable music based on users' mood in previous period and current emotion stimulus. A care factor was used to adjust the results so that users' mental status could be improved by the recommendation [46]. A real-time system that can recognize human faces, assess human emotions, and even recommend music to users. This system deploys deep learning-based CNN model, it can predict six emotions: anger, fear, joy, neutral, sadness, and surprise. The proposed system can be utilized in different places where real-time facial recognition plays an important role. It has achieved accuracy of 73.02% [47].

Table 1: State of the art techniques for music recommendation system

Sr. No.	Theme	Year of Publish	Methodology	Results	Dataset
1	MSR with Dynamic K-means Clustering Algorithm	2007	● Shortest Time Fourier Form ● K-means Clustering Algorithm	● Avg. Rmax: 0.5 ● Avg. cell size: 0.2 ● Accuracy: 80% ● MSR recommended mainly rock and classical music	Database consisted of 100 pieces each from: Jazz, ballad, classical, rock
2	MSR using Music Data Grouping and User Interest	2010	● K-means Algorithm	● 83% accuracy	100 MIDI files
3	DJ-Running MSR	2019	● Nearest Neighbor Search Algorithm ● Kubernetes Tech	Songs are selected considering the runner's profile, location and emotional state	Spotify
4	MSR using Classification	2019	● Classification on basis of Emotions and Genre ● AdaBoost Algorithm	Accuracy of: ● Five Genre Classification: 92.2% ● Four-emotion Classification: 90.5%	● Genre Dataset (1000 songs) ● Emotion Dataset (800 songs)
5	Content-based Music Recommendation System	2021	● Acoustic Feature Analysis ● DL and Computer Vision ● Euclidean, Manhattan, Cosine Distances	Precision: ● Random recommendation: 0.006 ● Genre-specific: 0.066 Acoustic analysis: 0.112	● MSD (Million Songs Dataset) ● FMA (Free Music Archive): 106,000 tracks

6	MSR for Emotion Detection using Facial Expression	2020	• Music Extraction Module • Audio Extraction Module • Emotion-audio integration Module	Accuracy of 80%	• Cohn Kanade Extended • HELEN
7	EmuPlayer: Based on User Emotion	2010	• Bio-signal Data of User • RF-ECG sensor	Accuracy: • Liked Songs: 64.5% • Disliked songs: 6.66% • Bad-influence: 8.57% Avg. satisfaction scale: 3.6/5	• MY SQL database was used to study user's emotions and mental state
8	FOAF & RSS Music Recommendation System	2005	• XML/RDF files • TF-IDF • Collaborative Filtering • Content-based Filtering	System's accuracy: 71.2%	• MP3 Blogs • Podcast sessions
9	MusiCube: Visual MRS	2011	• iGA • PCA	Colour Representation in 2-D plane: • Positive: Red • Negative: Orange Not detected: Yellow	• RWC Music Database
10	Next One Player	2011	• ARIMA • Gaussian Mixture Model • Forgetting Curve	System was able to fit a user's taste and adjust recommendation strategy quickly whenever user skips a song	• AllMusic • ID3v1 or ID3v2
11	T-RECSYS: MRS using Deep Learning	2019	• Google Tensorflow • Python library Kreas • Content Based Filtering • Collaborative Filtering	• High accuracy Readily extensible to different market services like iTunes, Amazon	Modified Spotify Dataset
12	TICI: User-Interests Approach to Music Recommendation	2011	Content Based method + Rank of the group weight	More accurate that Content Based Method	User Behaviour and emotion
13	TICI: MSR based on Music Data Grouping	2011	• Collaborative method • Density-Interest	COL method is more accurate than CB with help of DI method	User's interest
14	MSR based on Music Emotion Classification	2017	• AdaBoost Algorithm • Modified Brute-Force method	Recall rate and search speed are better than brute force method	2000 songs (4emotions x 500 songs each)
15	MSR based on User behaviour in Time Slot	2009	• Decision Tree Classification • C# • Flash	Overall precision of system is 78.33%	Database developed using Microsoft Access 2003
16	Multi-view enhanced graph attention network for session-based music recommendation	2023	• Graph Neural Network • Heterogeneous Music Graph	MEAGAN achieves better Precision, Recall, F1, Mean Average Precision in session-based music recommendation tasks.	Real world datasets: Lastfm and 30Music
17	Emotion based personalized MRS	2023	• LSTM model • Care factor	Precision 0.83 Recall 4.17 MSE 0.032 RMSE 0.178 MAE 0.291	GTZAN dataset
18	Facial emotion recognition and MRS using CNN	2024	• CNN • Deep Learning	Accuracy 73.02%	OAHEGA and FER-2013 datasets

3 Proposed methodology

Content based MRS makes analysis of the content of every song the user has heard in history, revealing a common rule that essentially restores a user's listening behavior. Songs that match this rule are recommended. In this way, content based MRS can recommend songs which perfectly match the listening profile of the user. Unlike a collaborative filtering method, a content-based approach enables predicting by analyzing song tracks [28, 31]. It is based on retrieving information and filtering information [32] recommending a song similar to the one the user has listened to in the past than the user has rated it as 'like' [33, 34]. Many studies focus on extracting and comparing acoustic features in finding tracks of common understanding [35, 36]. Most represented so far are rhythm [37, 38, 39]. Basing on features which were extracted, the distance lying between tracks is calculated [40]. Some standard calculations of similar k-methods that combine the distance of the earth converter, the increase in expectation by the sample of Monte Carlo and the average vectors in the Euclidean range. [41]. As per the dataset parameters, we will be using expectation maximization technique for content-based filtering uses sampled vectors from parameters of the two comparable songs; the sampling is performed via generation of weights [42]. Here mathematical algorithms like cosine similarity, TF-IDF will be used. Eventually the songs will be recommended on the basis of their allotted

personalized score according to the proposed methodology.

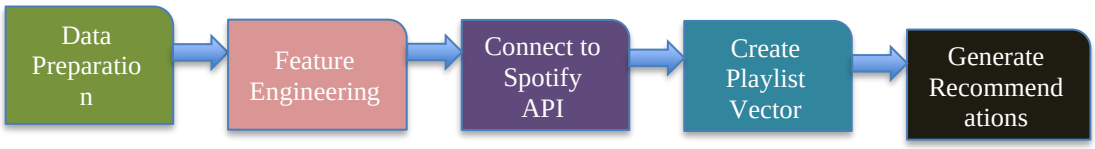


Figure 1: Content-Based filtering methodology

The steps to create recommendations are data wrangling, then creating workflows, connecting to the Spotify API, creating playlist vectors, and finally creating recommendations for users. The first step is the data preprocessing step that will be performed using a Python library called pandas. In this step, you can find comparison among the dance number, tone, life, voice, etc. of various opening songs has been done. Later we will be able to distinguish one of the names according to the type of music. The next step is to use feature engineering in conjunction with flexible variable customization to create hot code features and popular variants of the year. TF-IDF (Time Frequency and Inverse Document Frequency) properties for the actors are also created in this step. TF-IDF automatically displays metadata words according to their frequency (i.e. weight) in the entire list. The TFIDF diagram is shown in Figure 2 and Figure 3. The Figure 4 shows how to use TF-IDF in music application. The TF-IDF function creates a number vector of all songs by assigning a number value to each word in the word. The next step is to connect to Spotify API to download a specific playlist.

TF-IDF can be defined as a numerical statistic which indicates the significance of a word in given document. TF stands for term frequency while IDF stands inverse document frequency. The value of TF-IDF is directly proportional to the number of times a word occurs in the document and is offset by the frequency of the word in the corpus.

$$\boldsymbol{TF - IDF(t,r) = TF(t,r) * IDF(t)} \quad (1)$$

$$IDF(t) = log\frac{1+n_r}{1+df(r,t)} + 1 \quad (2)$$

When TF (t, r) indicates the number of times the word t appears in a review r, nr is the total number of revisions and the df (r, t) update number containing the word t. Reviews are considered the equivalent of a document. [27].

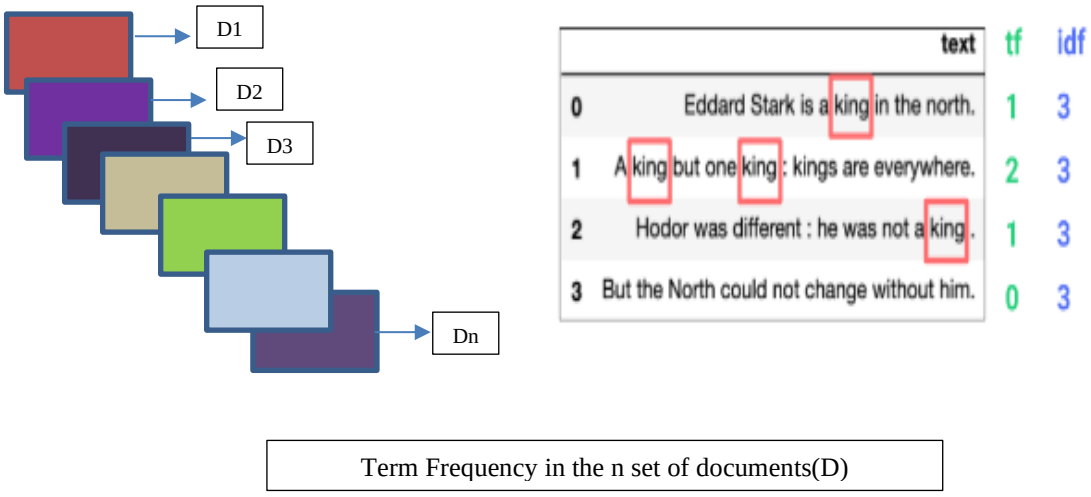


Figure 2: Pictorial representation to TF – IDF

$$cos\ (\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad \ldots\ldots. (3)$$

Eqn. (3) justifies the cosine similarity algorithm mathematically where A and B are vectors and Where $\vec{a}.\ \vec{b}. = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + a_3 b_3 + \cdots \ldots a_n b_n$ is the dot product of the two vectors.

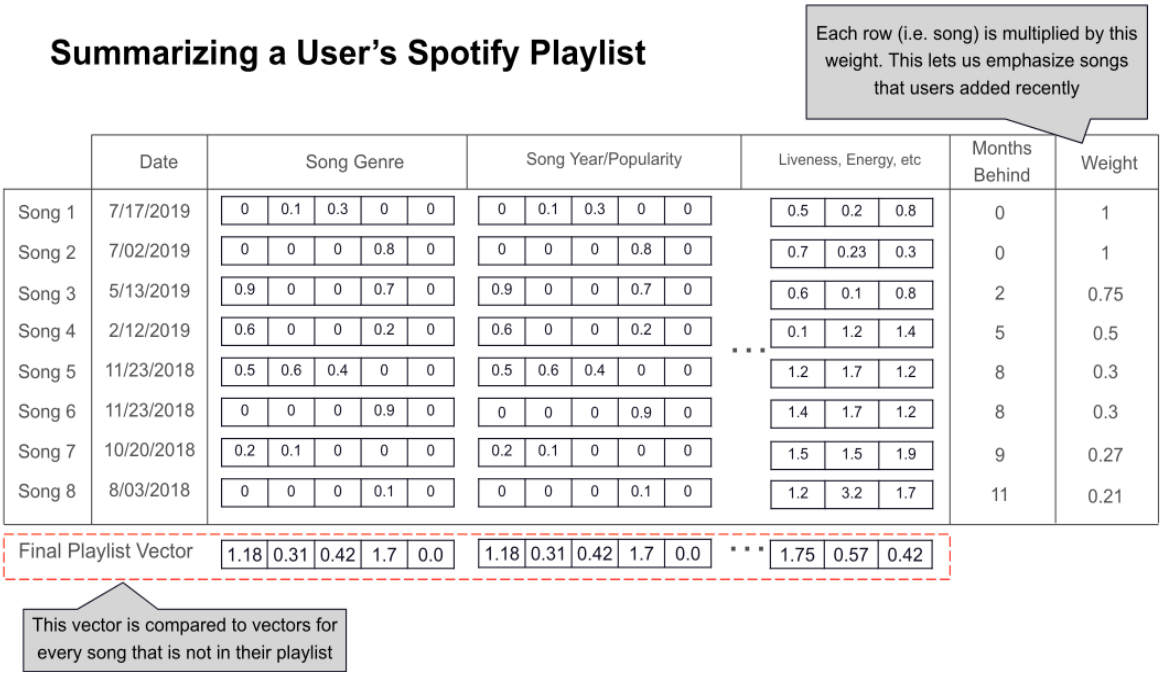


Figure 3: Summarizing a user's spotify playlist

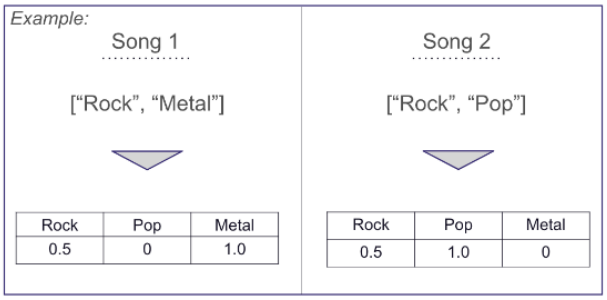


Figure 4: Example of TF – IDF

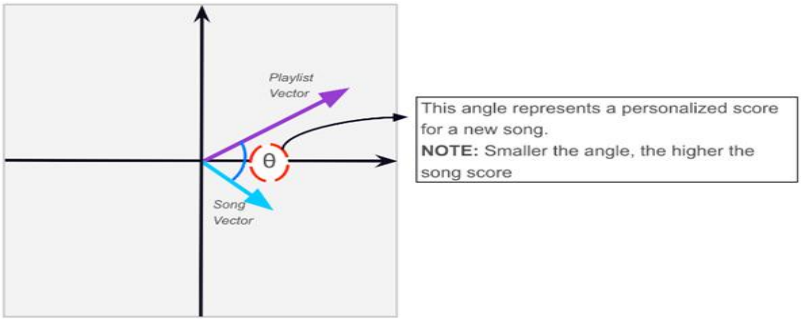


Figure 5: Cosine similarity

The playlist vector is produced by repeating each line namely, song, thunder year song and important things like

life, dancing etc. by shared weight. Figure 5.3 represents a summary of the User's Playlist, which includes columns like Song name (with the ID), Date, Song Genre, Popularity, All Feature Variables, Months Behind, and Weight. 'Date' represents the release date of the song, the song genre specifies how much exact percentage of each genre a song comprises. Nowadays, many artists tend to merge a number of genres together, and only a few artists are true to just one music style/genre. This can be explained in detail in figure 5.4. If we consider 2 song choices, the first song, Song 1, say, is put under the genre Metal, but it does not mean that the song is entirely Metal, Metal itself is a subdivision of Rock. Rock itself comprises several songs, but when it comes to putting the songs under categories, it gets categorized into subcategories. Here, Song 1, will have both rock and metal properties, but no Pop elements. Hence, while calculating the Final Playlist Vector, we will take into account only Rock and Metal. Similarly, Song 2, has some Rock and Pop elements to it, but no Metal elements. Popularity is given in the form of a gradient where the value ranges from 1 to 100, where 1 signifies the least popularity score, and 100 is very popular and likely is on the billboard charts. However, these columns are all that were in the original dataset as well, and the only new columns are Months Behind and Weight. The 'Months Behind' column signifies "When was the song added in the playlist" if the song was added in the same month, it's given a weight 0, if it has been 2 months, the weight becomes 0.75. The weight simply means recency bias and has a float value. The larger the recency bias value, the more priority the song gets. This is due to the reason that we need to focus on the user's most recent taste. We need to consider what the user likes most nowadays and not something that is outdated for them. The recommendations need to bias toward the new songs that attract users' attention. We needed songs that reflected the user's current/recent taste, hence considered weight. In the end, we just add up all the columns and get a final playlist vector, and this is where the 'Weight' actually comes into play. Since the weight is multiplied across all the rows, the last song, Song 8, will not have much significance while generating recommendations, since it has the lowest weight and hence will be affecting the sum as well. Cosine similarity helps in generating usic recommendations in the last step. There are more than 2600 inputs that go into this model, but here, we need to imagine that all those inputs are one arrow, since these inputs are just vectors. Each row of our feature would be considered a single vector. In Fig 5, both the arrows are vectors and have the same attributes, one represents the playlist of the user and the other represents the song. What cosine similarity actually does is that it takes the angle between both of these arrows/vectors. The angle represents how good of a recommendation that song is. The smaller the angle is, the higher the song score is. If the song vector and the playlist vector are pointing in the same direction at a very small angle, it means that the song is a good fit for the playlist. And that's how the recommendations are made.

3.1 Collaborative-based filtering methodology & experimental setup

This approach involves clustering users based on their preferences, facilitating the sharing and refinement of music among users within the same group. The filtering process typically relies on analyzing the content of genres, artists, or albums extracted from users' listening and download histories. One significant advantage of this method is the high likelihood that users will discover unexpected yet familiar songs through the recommendations. [8]. While collaborative filtering encompasses three categories—model-based, memory-based, and hybrid collaborative—we focus on model-based collaborative filtering in this context. This approach utilizes data mining and machine learning algorithms to train the system according to other users' preferences. It essentially represents user preferences using a set of rating scales and constructs a unique prediction model for comparison with other users' preferences. [28]. The system generates test forecasts and real-world data based on a known model. One of the primary challenges is generating online recommendations within a reasonable timeframe while handling large datasets. KNN (K-Nearest Neighbors) stands out as one of the most effective neighboring algorithms for collaborative filtering with large datasets. This algorithm considers the central ratings of each user, often referred to as a pivot [29]. By comparing this pivot value, the algorithm initiates the search for the most similar music, ultimately recommending the one with the highest likelihood estimation to the user [30].

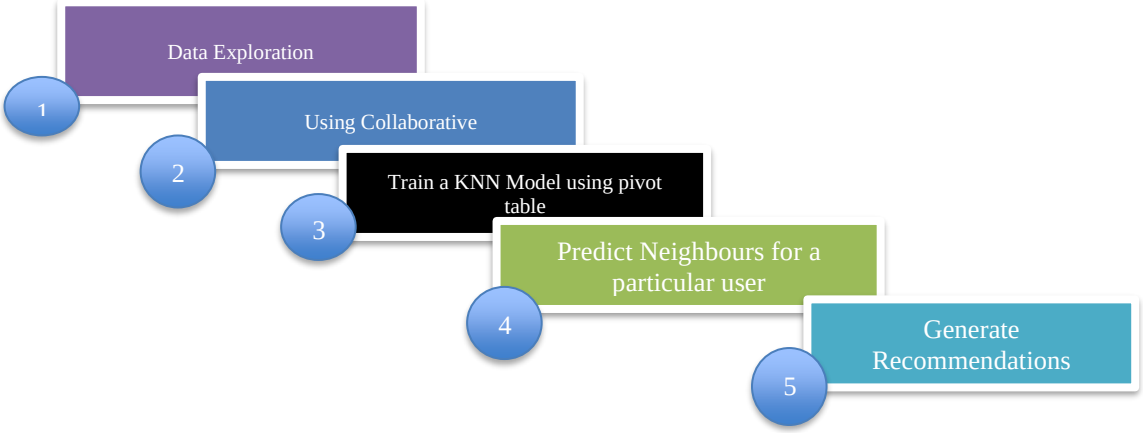


Figure 6: Collaborative-Based filtering methodology

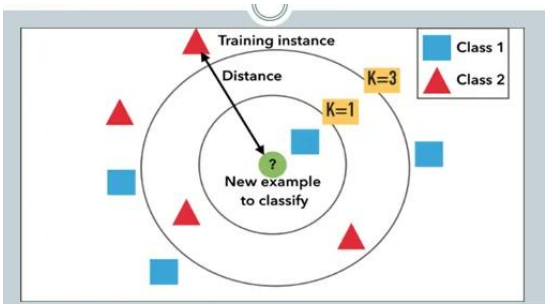


Fig 7: KNN algorithm decision process[1]

The prediction r_{ui} is set as:

$$r_{ui} = \frac{\sum_{v \in N_i^k(u)} sim(u,v) * r_{vi}}{\sum_{v \in N_i^k(u)} sim(u,v)} \quad (4)$$

Table 2: Parameter's measure

Parameters	**Meaning**
r_{ui}	The estimated rating of user u for item i.
$N_u^k(u)$	The *k* nearest neighbors of user *u* that have rated item *i.*
r_{vi}	The true rating of user *v* for item *i.*
Sim_options(dict)	A dictionary of options for the similarity measure.
k(int)	The number of neighbors to take into account for aggregation.

The meaning of each parameter in the formula (4) is shown in Table 2.
It can be seen from the formula that KNN is influenced by the ratings of the song given by other users as shown in Figure 6 and 7.

3.2 Dataset description

The user's Spotify Playlist is extracted using API calls, where the playlist includes a set of songs added by the user over a period, ranging from 0 to 12 months, i.e., a course of a year. In order to extract information about these songs, another dataset was used, which was available on Spotify's research website. The dataset itself consists of a set of a million playlists, including titles of those playlists, as well as the songs. It includes information about all the songs, including Song ID, Track, Artist Name, rhythm, vitality, pitch, volume, manner, vocal presence, acoustic quality, instrumental presence, liveliness, emotional tone, pace, song's time signature, date of release, and so forth.
We have considered a personalized playlist where the songs are not only the mainstream sellouts but also the less popular, yet appreciable music pieces. The playlists are customizable meaning, the songs can be added, deleted, or rearranged at any point in time. All the songs have the (approximate) same variable scores (example, valence, loudness, danceability, etc.). For instance, the songs "Body" and "Working for It", possess identical danceability scores of 0.752 and 0.776 respectively, and comparable energies as well. However, the songs do seem to be of fairly different keys when compared. However, the majority of songs are in key 11, popularly called key Bb (B flat). There are 12 keys in music: C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb, B. As for the time signature, every song has a time signature of 4. Apart from 2 outliers, the entire dataset seems to be very consistent.

[1]https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a

The personalized playlist, however, includes the name of the song, artist name, and date added. There are a total of 16 songs in the playlist, making the total duration 54:06 minutes. The songs added are by different artists to avoid a biased generation of recommendations. However, the time signature and mode of the songs are taken the same, 4 and 1, respectively.
There were minimal differences between the values of the variables, making it a consistent playlist, with minimal

outliers.

We visualized the cover arts of the respective songs that are present in the inputted dataframe using matplotlib. The column numbers were specified (here, 5) in order to present the cover arts along with the song names in a structured manner. Spaces between the covers were adjusted as well to avoid cluttering and provide enough space for the longer title names. The visual representation of playlist and set of playlists is shown Figure 8 and Table 3 respectively.

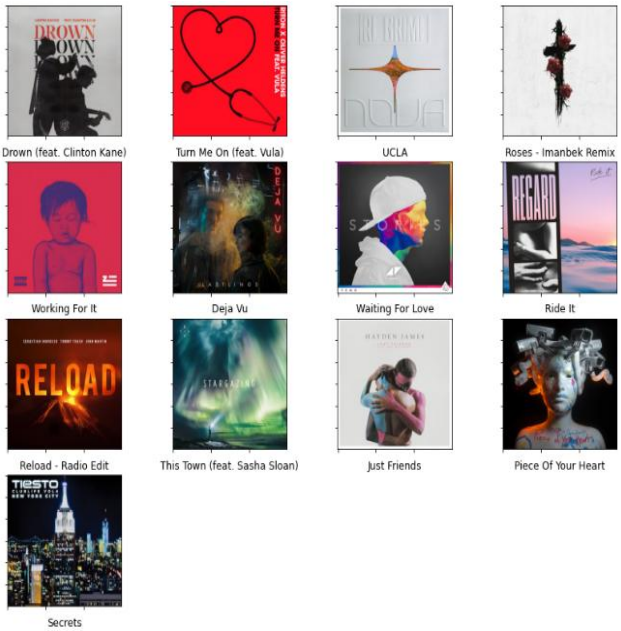


Figure 8: Visual representation of songs in our playlis

4 Experimental analysis and result

Upon analyzing the results of content-based recommendations, it was observed that the variables showed strikingly closer value to the songs in the original playlist. Taking an example of the recommendation, the song “Roots”, as generated, has a very close value of danceability, valence, tempo, energy, and key when compared to an average score of the original playlist table. The observed average variables were: Danceability: 0.67093, Energy: 0.75593, Key: 8, Loudness: -5.29927, Mode: 0, Speechiness: 0.06437, Acousticness: 0.11175, Instrumentalness: 0.07404, Liveness: 0.20707, valence: 0.51013, and Tempo: 124.56. Majority of the songs maintained danceability, energy, loudness, speechiness, and tempo, which can be seen as a major contributor and in the recommendations. Similarly, if we consider the song “Hold On (feat. Cheat Codes) - 2020 Edit”, we can see that the tempo is 124.859, which is very close to 124.5601, a liveness of 0.2630, and a loudness of -4.969, all of which are comparable to the ideal dataset values. However, there were major variations in mode (some songs were in major key some were in a minor key) like the song “No Money” and “Runaway (U & I)”, both being in major key, while all the songs in the dataset were in a minor key, and valence. These are Spotify’s Original Recommendations.

Initially, Spotify provides only 10 recommendations. Upon pressing the “Refresh” button provided, more songs can be recommended. But the limitation is that the songs will get repeated in the recommendations, and this step is highly influenced by the label the artists are under, and the popular songs in the country or worldwide at the moment. Hence, these are not true to the users’ taste. Along with this, when compared to the songs in the playlist, major deviances were observed, especially in Acousticness, Key, Mode, and Valence. However, there was a match in the time signature, except for 2 to 3 outliers. Upon analyzing the results of collaborative filtering, it was observed that the majority of the song recommendations were similar to Spotify's original recommendations, mainly the songs by Galantis and Steve Aoki. It was also observed that the songs were heavily influenced by the playlists of other users and the user’s overall preference. In this case, the user's top artists for the past 2 years were Tove Lo, Troye Sivan, and Panic! At the Disco. Hence, the recommendations were biased towards them, however, the recommendations were the remixes of the popular songs by these artists, since they fit the overall genre better as compared to original songs which have a significantly divergent tempo, valence, energy, and danceability. All the remixes that are recommended have done extremely well on the billboards in the past. Examples being One Kiss by Dua Lipa, Copycat by Billie Eilish, Attention by Charlie Puth, and Talking body by Tove Lo, Without Me by Halsey, to name a few. The details are shown in Figure 9, 10 and 11. The result analysis has been performed on 2 category of filtering techniques (Content based and collaborative filtering technique) as shown in Table 4. The Table 4 clearly shows that content-based filtering approach outperform earlier work.

Table 3: Our playlist

Artist	**Name**	**Id**	**url**	**Date_added**
Martin Garrix	Drown (feat. Clinton Kane)	4RVtBIHFKj51lpvpfv5ER4	https://i.scdn.co/image/ab67616d00001e02b154bc...	2020-08-01 01:27:34+00:00
Riton	Turn Me On (feat. Vula)	OqaWEvPkts34WF68r8Dzx9	https://i.scdn.co/image/ab67616d00001e02216a27.	2020-07-09 :34:03+00:00
RL Grime	UCLA	3OaunNUIXXs5e2PXtNAzzG	https://i.scdn.co/image/ab67616d00001e02eded2e...	2020-06-20 00:34:44-00:00
SAINt JHN	Roses - Imanbek Remix	7fPuWripwDcHm5aHCH5D9t	https://i.scdn.co/image/ab67616d00001e022b6e2f.	2020-04-19 06:26:21+00:00
Loud Luxury	Body	21RzyxY3EFaxVy6K4RqaU9	https://i.scdn.co/image/ab67616d00001e02af5e18...	2020-03-26 22:28:23+00:00

ZHU	Working For It	2HJQcyUpmUuvzS5vBAICIc	https://i.scdn.co/image/ab67616d00001e02bfaac9...	2019-12-19 15:53:47+00:00
Lastlings	Deja Vu	649HM5IOHHqsoG5nldMo6L	https://i.scdn.co/image/ab67616d00001e02129817.	2019-11-19 16:04:48+00:00
Avicii	Waiting For Love	2P4OICZRVAQcYAV2JReRfj	https://i.scdn.co/image/ab67616d00001e025393c5...	019-11-17 03:38:47+00:00
Regard	Ride It	2tnVG71enUj33lc2nFN6kZ	https://i.scdn.co/image/ab67616d00001e025c2781...	2019-11-13 04:13:21+00:00
Dimitri Vegas & Like Mike	Mammoth	76fqWMe0buqQoaNTIbLWmr	https://i.scdn.co/image/ab67616d00001e0216bf35...	2019-10-26 19:11:43+00:00
Sebastian Ingrosso	Reload Radio Edit	5jyUBKpmaH670zrXrEOwmO	https://i.scdn.co/image/ab67616d00001e0270e2e5...	2019-10-04 15:50:31+00:00
Kygo	This Town (feat. Sasha Sloan)	4aSfgWmRa9KsISD4Jmx7QB	https://i.scdn.co/image/ab67616d00001e02a33355.	2019-09-30 20:05:19+00:00
Hayden James	Just Friends	6tB4XVKceo2307SSWXaO0y	https://i.scdn.co/image/ab67616d00001e024b6940.	2019-09-30 20:04:53+00:00
MEDUZA	Piece Of Your Heart	1DFD5Fotzgn6yYXkYsKiGs	https://i.scdn.co/image/ab67616d00001e02ead130...	2019-09-30 20:04:47+00:00
Dimitri Vegas & Like Mike	Tremor - Sensation 2014 Anthem	6AE0G24YXnDyEgE4L0efpB	https://i.scdn.co/image/ab67616d00001e023d4c4f.	2019-09-30 20:04:43+00:00
Tiesto	Secrets	ONIC4unbe5KZOp1d9T7OaF	https://i.scdn.co/image/ab67616d00001e02de5f51...	2019-09-30 20:04:40+00:00

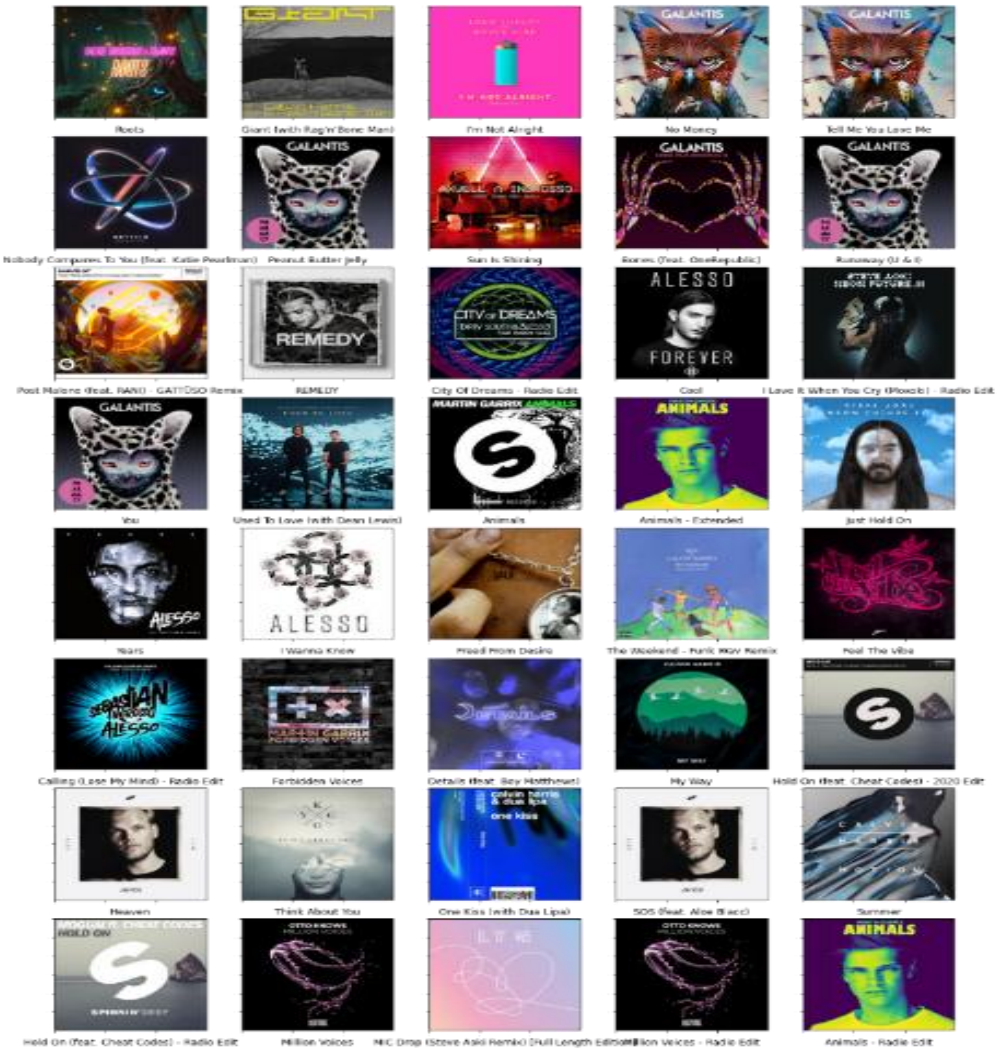


Figure 9: Content-Based recommendations

Table 4: Comparative result study of proposed work

Filtering Technique	Accuracy	Precision	Recall
Content based (Proposed work)	0.85	0.84	0.81
Collaborative based (Proposed work)	0.81	0.80	0.79
Earlier work	0.71	0.71	0.70

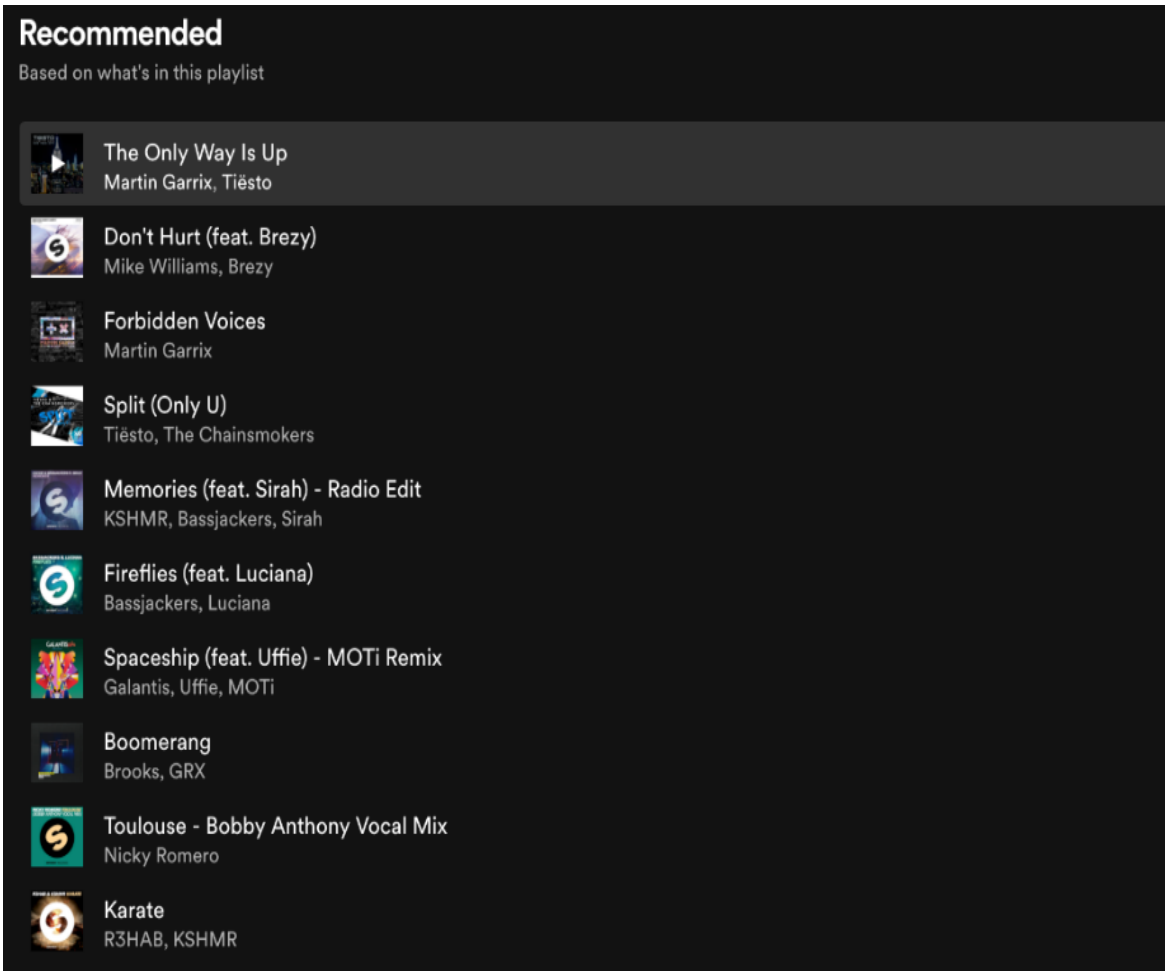


Figure 10: Spotify recommendations

4 Conclusion and future work

In this paper, we have proposed a novel approach to building a content based filtering and collaborative based music recommendation system using mathematical algorithms and KNN. The result of the Content-Based Recommendations was much closer to the user's initial taste than collaborative filtering, upon comparing the attributes, it was found that all variables were remarkably closer to the ones in the source playlist, except the time signature. . From the experimental analysis in has been identified that content-based filtering technique performed best on KNN machine learning classifier with accuracy of 85%.

Unlike Spotify's recommendation system which emphasizes mostly on valence, the proposed recommendation systems recommended songs with parameters like tempo, liveliness, danceability and loudness being in comparable range with those of the user's original playlist songs while keys used and mode observed high variations. Hence, recommendation systems should focus on more than just valence in order

to recommend better and personalized music to listeners. The future work of this project includes implementation of a recommendation system based on deep learning in order to generate more accurate and user specific music recommendations for both content and collaborative based filtering.

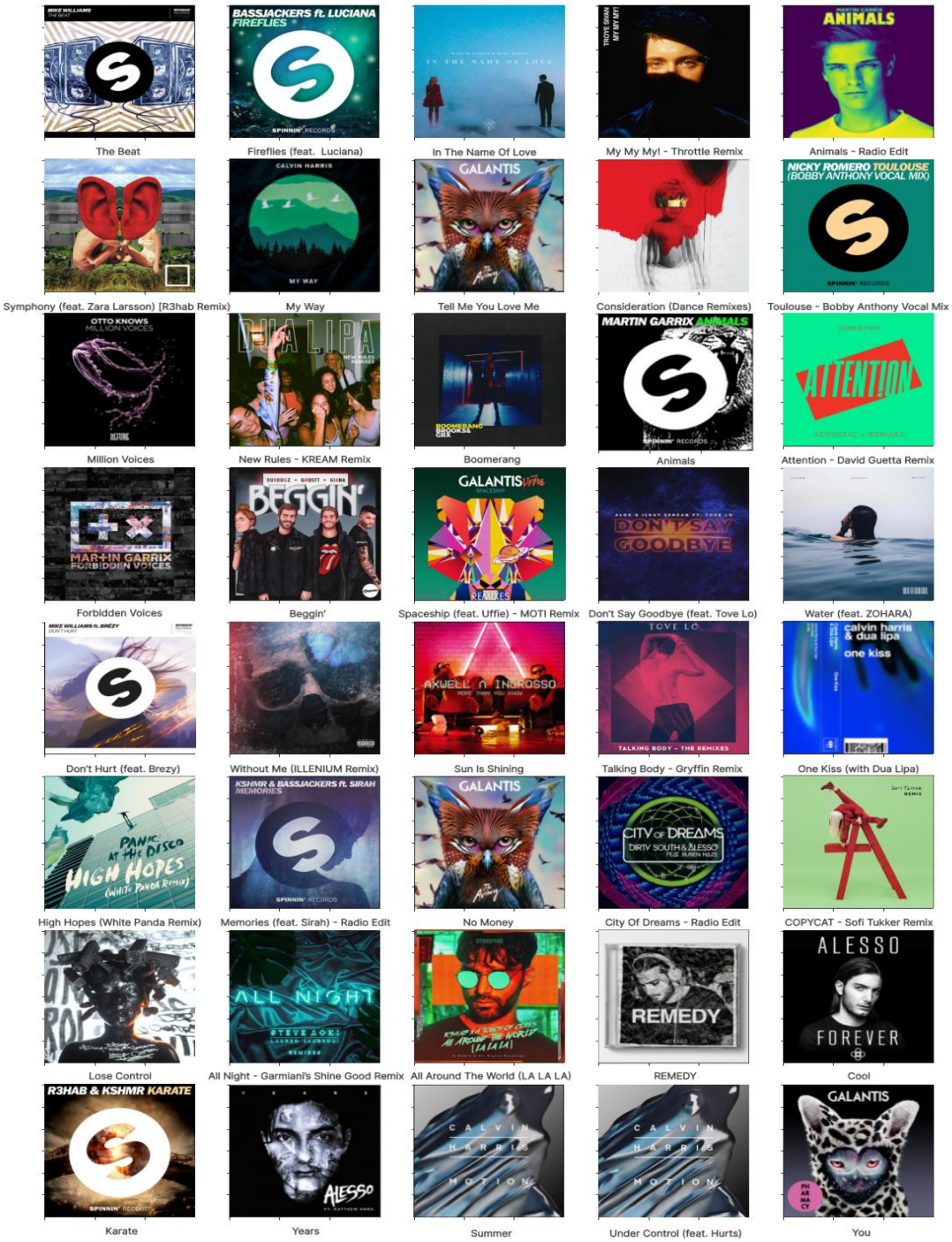


Figure 11: Collaborative based recommendations

References

[1] D. Kim, K. Kim, K. Park, J. Lee and K. M. Lee, "A music recommendation system with a dynamic k-means clustering algorithm," Sixth International Conference on Machine Learning and Applications (ICMLA 2007), 2007, pp. 399-403, doi: 10.1109/ICMLA.2007.97.

[2] Chen HC, Chen AL. A music recommendation system based on music data grouping and user interests. InProceedings of the tenth international conference on Information and knowledge management 2001 Oct 5 (pp. 231-238).

[3] Song Y, Dixon S, Pearce M. A survey of music recommendation systems and future perspectives. In9th international symposium on computer music modeling and retrieval 2012 Jun 19 (Vol. 4, pp. 395-410).

[4] Schedl M. Deep learning in music recommendation systems. Frontiers in Applied Mathematics and Statistics. 2019:44.

[5] Zhu X, Shi YY, Kim HG, Eom KW. An integrated music recommendation system. IEEE Transactions on Consumer Electronics. 2006 Oct 9;52(3):917-25.

[6] Niyazov A, Mikhailova E, Egorova O. Content-based music recommendation system. In2021 29th Conference of Open Innovations Association (FRUCT) 2021 May 12 (pp. 274-279). IEEE.

[7] Florence SM, Uma M. Emotional Detection and Music Recommendation System based on User Facial Expression. InIOP Conference Series: Materials Science and Engineering 2020 Aug 1 (Vol. 912, No. 6, p. 062007). IOP Publishing.

[8] Le NT. EmuPlayer: Music Recommendation System Based on User Emotion Using Vital-sensor. Bachelor Dissertation. Keio University. 2010.

[9] Celma Ò, Ramírez M, Herrera P. Foafing the Music: A Music Recommendation System based on RSS Feeds and User Preferences. InISMIR 2005 Sep 11 (pp. 464-467).

[10] Saito Y, Itoh T. MusiCube: a visual music recommendation system featuring interactive evolutionary computing. InProceedings of the 2011 Visual Information Communication-International Symposium 2011 Aug 4 (pp. 1-6).

[11] Hu Y, Ogihara M. NextOne Player: A Music Recommendation System Based on User Behavior. InISMIR 2011 Oct (Vol. 11, pp. 103-108).

[12] Fessahaye F, Perez L, Zhan T, Zhang R, Fossier C, Markarian R, Chiu C, Zhan J, Gewali L, Oh P. T-recsys: A novel music recommendation system using deep learning. In2019 IEEE international conference on consumer electronics (ICCE) 2019 Jan 11 (pp. 1-6). IEEE.

[13] Chang YI, Wu CC, Tsai MC. A user-interests approach to music recommendation. InProceedings of the World Congress on Engineering 2011 (Vol. 3).

[14] Chang YI, Wu CC, Tsai MC. A Fair Approach to Music Recommendation Systems Based on Music Data Grouping. IAENG International Journal of Computer Science. 2011 Oct;38(4):418-27.

[15] Kim HG, Kim JH. Music Similarity Search Based on Music Emotion Classification. The Journal of the Acoustical Society of Korea. 2007;26(3E):69-73.

[16] Liu NH, Lai SW, Chen CY, Hsieh SJ. Adaptive music recommendation based on user behavior in time slot. IJCSNS International Journal of Computer Science and Network Security. 2009 Feb 28;9(2):219-27.

[17] Xing Z, Wang X, Wang Y. Enhancing Collaborative Filtering Music Recommendation by Balancing Exploration and Exploitation. InIsmir 2014 Oct 27 (pp. 445-450).

[18] Shakirova E. Collaborative filtering for music recommender system. In2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) 2017 Feb 1 (pp. 548-550). IEEE.

[19] Su JH, Chang WY, Tseng VS. Effective social content-based collaborative filtering for music recommendation. Intelligent Data Analysis. 2017 Jan 1;21(S1): S195-216.

[20] Hu R, Pu P. Enhancing collaborative filtering systems with personality information. InProceedings of the fifth ACM conference on Recommender systems 2011 Oct 23 (pp. 197-204).

[21] Van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. Advances in neural information processing systems. 2013;26.

[22] Shao B, Wang D, Li T, Ogihara M. Music recommendation based on acoustic features and user access patterns. IEEE Transactions on Audio, Speech, and Language Processing. 2009 Sep 1;17(8):1602-11.

[23] Crisan A, Fisher SE, Gardy JL, Munzner T. GEViTRec: Data Reconnaissance Through Recommendation Using a Domain-Specific Visualization Prevalence Design Space. IEEE Transactions on Visualization and Computer Graphics. 2021 Aug 27.

[24] Sen A, Larson MA. From Sensors to Songs: A Learning-Free Novel Music Recommendation System using Contextual Sensor Data. InLocalRec@ RecSys 2015 Sep 19 (pp. 40-43).

[25] P. Knees and M. Schedl, Music similarity and retrieval: an introduction to audio- and web-based strategies. Berlin: Springer, 2016.

[26] Yazhong Feng and Y Zhuang. Popular Music Retrieval by Detecting Mood. In International Society for Music Information Retrieval 2003, volume 2, pages 375– 376, 2003

[27] Vashishtha S, Susan S. Inferring sentiments from supervised classification of text and speech cues

using fuzzy rules. Procedia Computer Science. 2020 Jan 1; 167:1370-9.

[28] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6):734–749, June 2005.

[29] G. Li and J. Zhang, "Music personalized recommendation system based on improved KNN algorithm," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018, pp. 777-781, doi: 10.1109/IAEAC.2018.8577483.

[30] Liu NH. Comparison of content-based music recommendation using different distance estimation methods. Applied intelligence. 2013 Mar;38(2):160-74.

[31] Qing Li, Byeong Man Kim, Dong Hai Guan, and Duk Oh. A Music Recommender Based on Audio Features. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pages 532–533, Sheffield, United Kingdom, 2004. ACM.

[32] M.A. Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based Music Information Retrieval: Current Directions and Future Challenges. Proceedings of the IEEE, 96(4):668–696, 2008.

[33] Jean-julien Aucouturier and Francois Pachet. Music Similarity Measures: What is the Use. In Proceedings of the ISMIR, pages 157–163, 2002.

[34] Beth Logan. Music Recommendation from Song Sets. In International Conference on Music Information Retrieval 2004, number October, pages 10–14, Barcelona, Spain, 2004.

[35] Dmitry Bogdanov, J. Serra, Nicolas Wack, Perfecto Herrera, and Xavier Serra. Unifying Low-level and High-level Music Similarity Measures. IEEE Transactions on Multimedia, 13(99):1–1, 2011.

[36] Chun-man Mak, Tan Lee, Suman Senapati, Yu-ting Yeung, and Wang-kong Lam. Similarity Measures for Chinese Pop Music Based on Low-level Audio Signal Attributes. In 11th International Society for Music Information Retrieval Conference, number ISMIR, pages 513–518, 2010.

[37] Dmitry Bogdanov and Perfecto Herrera. How Much Metadata Do We Need in Music Recommendation? A Subjective Evaluation Using Preference Sets. In 12th International Society for Music Information Retrieval Conference, number ISMIR 2011, pages 97–102, 2011.

[38] Pedro Cano, Markus Koppenberger, and Nicolas Wack. An Industrial-strength Content-based Music Recommendation System. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’05, page 673, New York, New York, USA, 2005. ACM Press.

[39] Pedro Cano, Markus Koppenberger, and Nicolas Wack. Content-based Music Audio Recommendation. In Proceedings of the 13th annual ACM international conference on Multimedia, number ACM, pages 211–212, 2005.

[40] Beth Logan. Music Recommendation from Song Sets. In International Conference on Music Information Retrieval 2004, number October, pages 10–14, Barcelona, Spain, 2004.

[41] Terence Magno and Carl Sable. A Comparison of Signal of Signal-Based Music Recommendation to Genre Labels, Collaborative Filtering, Musicological Analysis, Human Recommendation and Random Baseline. In ISMIR 2008: proceedings of the 9th International Conference of Music Information Retrieval, pages 161–166, 2008.

[42] F. Pachet and J.J. Aucouturier. Improving Timbre Similarity: How High is the Sky? Journal of negative results in speech and audio sciences, 1(1):1–13, 2004.

[43] FlexerA, Stevens J. Mutual proximity graphs for improved reachability in music recommendation. Journal of new music research. 2018 Jan 1;47(1):17-28.

[44] Sarin, E., Vashishtha, S., & Kaur, S. (2022, February). SentiSpotMusic: a music recommendation system based on sentiment analysis. In 2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST) (pp. 373-378). IEEE.

[45] Wang, D., Zhang, X., Yin, Y., Yu, D., Xu, G., & Deng, S. (2023). Multi-view enhanced graph attention network for session-based music recommendation. *ACM Transactions on Information Systems*, *42*(1), 1-30.

[46] Liu, Z., Xu, W., Zhang, W., & Jiang, Q. (2023). An emotion-based personalized music recommendation framework for emotion improvement. *Information Processing & Management*, *60*(3), 103256.

[47] Bakariya, B., Singh, A., Singh, H., Raju, P., Rajpoot, R., & Mohbey, K. K. (2024). Facial emotion recognition and music recommendation system using CNN-based deep learning techniques. *Evolving Systems*, *15*(2), 641-658.

An Efficient Transferred Cascade System for COVID-19 Detection from Chest X-ray Images

Nassima Dif[1] and Zakaria Elberrichi[2]
[1]Ecole Superieure en Informatique, LabRI-SBA Lab, Sidi Bel Abbes, Algeria
[2]Computer science Department, Djillali Liabes University, EEDIS Laboratory, Sidi Bel Abbes, Algeria
E-mail: n.dif@esi-sba.dz, elberrichi@gmail.com

Keywords: Covid-19, classification, cascade models, chest x-ray images

Received: March 16, 2024

Analysing x-ray images for detecting Covid-19 presents one cost-effective approach. To automate this task, deep learning techniques have been suggested to reduce doctors workload. However, existing datasets classify X-ray images into three categories: Normal, Pneumonia, and COVID-19, but it is crucial to differentiate between bacterial and viral pneumonia due to their distinct treatment approaches. This paper introduces three novel cascade systems designed to distinguish between COVID-19 and non-COVID-19 pneumonia, as well as to classify bacterial and viral pneumonia, using a newly compiled dataset. The proposed Transferred Cascade Convolutional Neural Network (TCCNN) system enables the model to efficiently recognize complex concepts by combining various convolutional neural networks in two or three stages. Furthermore, TCCNN incorporates transfer learning within the cascade structure, allowing each convolutional neural network to exploit the trained model from the previous stage. The comparative analysis demonstrated the efficiency of the proposed systems, where the two-stage PN_CBV system achievd an accuracy of 96.27% using the DenseNet201_DenseNet121 combination.

Povzetek: Prispevek predstavlja nov pristop za prepoznavanje COVID-19 iz rentgenskih posnetkov prsnega koša z uporabo prenesenega kaskadnega sistema (TCCNN). Sistem razvršča slike v štiri kategorije: normalno, bakterijsko pljučnico, virusno pljučnico in COVID-19 s pomočjo prenesenega učenja.

1 Introduction

Since December 2019, new pneumonia has been detected in China and has affected a large number of people. This disease had a similar behaviour as severe acute respiratory syndrome (SARS). In March 2020, the World Health organization (WHO) declared this pandemic as COVID-19 [1]. The symptoms of COVID-19 include fever, cold, dry cough, breathing difficulties, and acute respiratory syndrome [2]. Because of its high transmissibility, controlling the spread of the virus has become urgent.

For the diagnosis of COVID-19, there are three main clinical tools in use: Real-time polymerase chain reaction (RT-PCR), computerized chest tomography (CT), and chest X-Rays (CXR) scans. RT-PCR tests risk missing positive cases due to various technical problems. Moreover, testing kits and the long processing time (4–6 hours [3]) can result in a rapid spreading rate of COVID-19. As an alternative and to control the Covid-19 spreading, radiological images such as X-rays and CT-scans have been exploited. While CT imaging presents several disadvantages, such as high radiation doses and sensitivity to patient movements [4], X-ray imaging is patient-friendly, fast, cheap, and can detect the disease early. Additionally, X-ray scanners are largely available. However, X-ray scans take a long time to detect COVID-19 and require an expert radiologist. Moreover, their manual analysis is time-intensive and can be influenced by doctors' subjectivity. To reduce the doctor's workload and avoid their subjective decisions, researchers have proposed exploiting computer-aided diagnostic systems (CAD) for COVID-19 detection from X-rays.

Recent advances in machine learning (ML) and deep learning (DL) techniques have enabled the development of CAD systems for X-ray image analysis [5, 6, 7, 8]. Convolutional neural networks (CNNs) are the most popular architectures due to their advantages for image analysis [9, 10, 11, 12]. However, these architectures risk overfitting due to the lack of data. To address this problem, many efforts have been made to create large datasets from different sources [13, 14, 15, 16]. To the best of the authors' knowledge, all voluminous datasets categorize X-rays into three classes: Covid-19, Normal, and Pneumonia due to the difficulty of multi-class classification systems.

The purpose of this contribution is to propose a four-classification system that distinguishes between COVID-19, normal, and viral and bacterial pneumonia. This classification helps to avoid misclassifying COVID-19 samples due to their similar characteristics with other viral pneumonia [17]. To simplify the multi-class classification task, we proposed a cascade system that divides the classification according to annotation complexity, starting with the easiest and progressing to the most complex. TCCNN helps

to quickly identify more complex concepts from data by stacking CNNs sequentially. The main contribution is that each model in our cascade system exploits the experience of the previous model through transfer learning. This was motivated by the advantages of transfer learning between similar and non-distant classification tasks.

This paper presents several contributions, including the proposition of three types of cascade systems based on six CNN architectures: two types of two-stage cascade strategies and a three-stage cascade system for covid-19 classification. Our study integrates a transfer learning strategy within the proposed systems.

The remaining parts of this paper proceed as follows: the first section details the related works to deep learning methods for COVID-19 classification. Section 2 explains the proposed method. Section 3 presents and discusses the obtained results. Finally, the last section concludes this work.

2 Related works

To address the COVID-19 epidemic, many efforts have been made to design deep learning applications for COVID-19 detection based on X-rays [18, 14, 19], and CT scans [20] of the chest. Various deep learning architectures have been exploited, such as CNNs [21] and Long short-term memory (LSTM) [22]. CNNs attracted much interest in detecting COVID-19 from X-rays due to their advantages for image processing. Predictive models for these architectures can be generated by either training from scratch [18] or using transfer learning techniques [14, 15, 19, 21]. Nayak et al. [21] highlighted the important results of transfer learning from the ResNet−50 compared to the other six CNN architectures. Chowdhury et al. [14] used a transfer learning technique from 7 imageNet models and one pre-trained on X-ray images (CheXNet). Their experimental study revealed that CheXNet was more efficient for binary classification, whereas, DenseNet201 was more promising in the three-class classification scheme. In another investigation [19], using the fine-tuned ResNet50 architecture as a feature extractor with SVM was more efficient than other pre-trained models and a trained CNN architecture from scratch. In [18], a new residual architecture was proposed to extract features at different abstraction levels. This approach uses two parallel convolutions with different filter sizes to capture multi-scale features. Another approach, suggested by Öksüza et al. [23], proposes using fine-grained, coarse-grained, and coarser-grained maps generated from three different networks: SqueezeNet, ShuffleNet, and EfficientNet-B0. This ensemble approach achieved promising results in detecting COVID-19 from chest x-ray images.

To combine the decisions of several learners, Win et al. [24] proposed an ensemble deep learning technique. Their approach was evaluated using 11 types of CNNs. The authors combined the five best models using soft and hard voting techniques. Similarly, Brunese et al. [25] proposed an approach that combines various VGG16 models in a three-fold binary classification framework. The first stage classified X-rays into normal or pulmonary cases, while the second stage differentiated COVID-19 from non-COVID-19 pneumonia. Finally, the third stage used the Gradient Class Activation Map (Grad-CAM) to localize suspected COVID-19 areas.

Our proposed contribution shares some similarities with the approach presented in [26], which proposed a two-cascade network. Their method first categorizes x-ray images into normal, pneumonia, or tuberculosis, and then distinguishes between normal pneumonia and covid-19 using the most efficient network for each level. In contrast, our contribution proposes three cascade strategies, including two types of two-stage cascade systems and a three-stage cascade strategy, with transfer learning applied within each stage. Furthermore, our approach has also the capability to classify other pneumonia diseases, such as bacterial and viral infections.

3 Proposed method

The purpose of cascade systems in COVID-19 detection from X-rays is to distinguish between normal and pneumonia classes or between pneumonia sub-classes at each level. In these systems, the classification is divided based on the complexity of annotations, starting from the easiest to the most complex. Discriminating between normal and pneumonia is less challenging than classifying pneumonia sub-classes, such as COVID-19, viral, and bacterial. Dividing the classification process into several stages reduces the model's classification load and can also help to reduce the error rate. These cascade systems are sequential ensemble learning strategies that combine the decisions of multiple learners to improve generalization. The originality of our ensemble learning system lies in the ability of each model to use the previous model's experience through a transfer learning method within the cascade system. Figure 1 illustrates the general structure of the proposed cascade system.

For the two-stage cascade system, X-ray images are classified into n categories using a pre-trained CNN on the ImageNet dataset. Subsequently, the first class is partitioned into m sub-categories, and the resulting dataset is passed through a second CNN. This network can be either a pre-trained CNN on ImageNet or the model developed in the first training stage. The purpose of transfer learning from stage 1 is to exploit the extracted features from dataset 1 to classify the sub-dataset which contains m sub-classes (class1.1, class1.2, ..., class1.m). The three-stage cascade strategy includes an additional step compared to the two-stage cascade method, where subclass 1.1 of class 1 is further divided into k sub-categories (class1.1.1, class1.1.2, ..., class1.1.k). Finally, the resulting dataset is passed through a pre-trained CNN from ImageNet or the previously fine-tuned model from the second stage, and the generated model is fine-tuned on this new dataset.

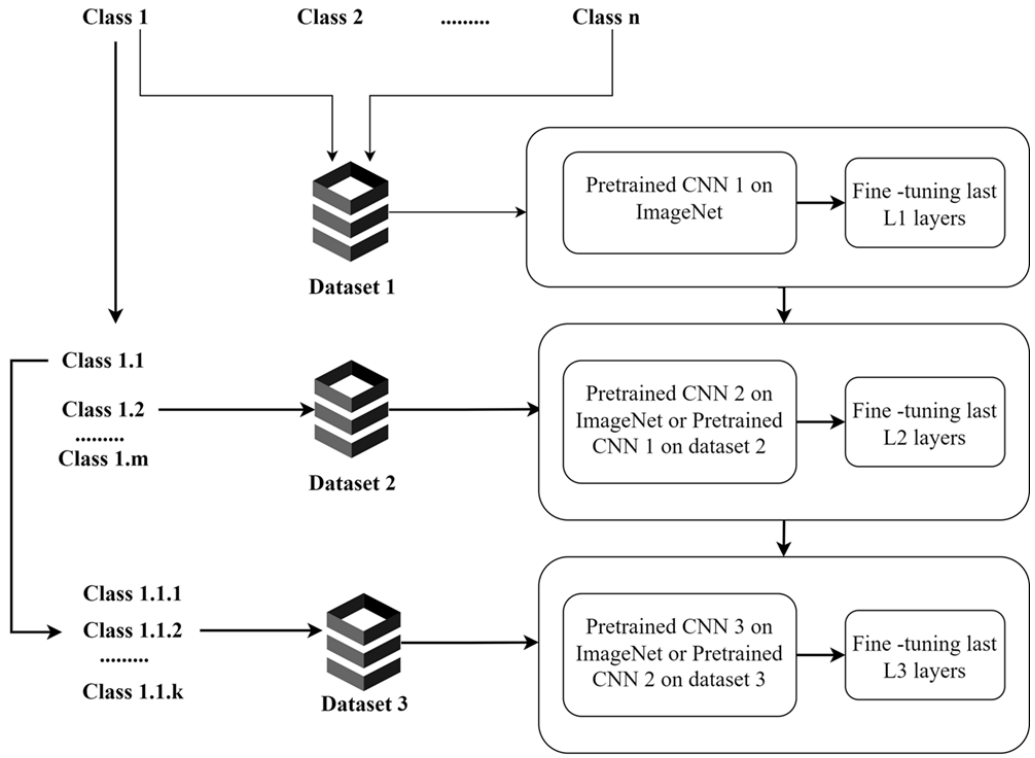


Figure 1: The proposed cascade strategy

The first two-stage scheme denoted as pneumonia COVID-19 normal_bacterial viral (PCN_BV), involves the classification of three distinct classes, namely non-COVID-19 pneumonia, COVID-19, and normal. Among these, the non-COVID-19 pneumonia class is further divided into a sub-group of m classes consisting of bacterial and viral categories. On the other hand, the second system pneumonia normal_COVID-19 bacterial viral (PN_CBV) deals with the classification of pneumonia and normal classes, and the group of pneumonia sub-classes consists of COVID-19, bacterial, and viral. The main difference between the two-stage schemes is the classification level of the COVID-19 class. The primary objective of this study is to determine whether it is challenging to distinguish between COVID-19, pneumonia, and normal classes in the first stage or to classify COVID-19, viral, and bacterial images in the second stage. In the three-stage cascade system pneumonia normal_pneumonia COVID-19_bacterial viral (PN_PC_BV), the group of n classes includes normal and pneumonia. The pneumonia class is further divided into non-pneumonia COVID-19 and COVID-19 subcategories. Finally, the last group of k classes presents subcategories of non-pneumonia COVID-19, which include bacterial and viral pneumonia.

In this study, we employed the transfer learning technique from ImageNet due to its extensive use in the literature and its efficiency. Additionally, we proposed to perform transfer learning between different models in the cascade system. This method aims to reuse previously extracted features by the CNN in the previous stage, as the label "i" in stage k groups images that share similar features and morphology to images in its subcategories in stage k+1. In the cascade system, the trained network on dataset k was used as a source model for transfer learning to dataset $k+1$. The transfer learning method consists of three main steps. First, the target CNN is initialized by the weights (P(X_s | Y_s)) of the source CNN, which was previously trained on ImageNet or X-ray images. Then, the last fully connected layers are removed and replaced by two fully connected layers and one softmax layer. The fully connected layers are composed of 1024 and 512 neurons, respectively, while the softmax layer contains C neurons, where C represents the number of labels in the target domain. Finally, the new network is fine-tuned on the target task. In this study, we used six CNN architectures: VGG16, VGG19, Inception, Xception, DenseNet201, DenseNet121. The fine-tuning process was performed according to CNN's nature.

4 Experimental study

The experiments were conducted on a computer with an Intel i5-core processor, 8 GB RAM, and NVIDIA GeForce GTX 1080 graphics processing unit (GPU), running on a 64-bit Ubuntu 16.04 operating system with Python.

We trained all CNNs based on the transfer learning strategy in 20 epochs with a batch size of 64. We used the Adam optimizer with a learning rate of 0.001. For evaluation, we used the stratified hold-out method: 60% for training, 20% for validation, and 20% for the test. To validate the efficiency of the proposed systems, we used four evaluation metrics, namely: accuracy, recall, precision, and F1-score.

4.1 Data compilation

The emergence of the COVID-19 pandemic has led researchers to propose voluminous datasets. To the best of the authors' knowledge, all voluminous datasets such as COVID-19 Radiography Database[1], COVID-QU-Ex [27], and COVIDx [2] classify CXR images into three classes: Normal, Pneumonia, and COVID-19. In this context, the pneumonia class groups viral, bacterial, and other pneumonia variants.

It is crucial to distinguish between bacterial and viral pneumonia since they require different forms of treatment. Bacterial pneumonia can be treated with antibiotics, while viral pneumonia requires supportive care [28]. Therefore, our study aims to distinguish between COVID-19 and non-COVID-19 pneumonia and classify bacterial and viral pneumonia. To generate our multi-source dataset COVID-QU-Ex_4C, we used three publicly available datasets: COVID-QU-Ex [3] [27], Chest-Xray-Pneumonia [4] [28], and Qata-cov19 [5]. Figure 2 highlights some samples from the COVID-QU-Ex_4C dataset.

Table 1 provides an overview of the proposed COVID-QU-Ex_4C dataset. The dataset includes normal and COVID-19 samples selected from the COVID-QU-Ex dataset, as well as bacterial and viral classes obtained by merging bacterial and viral images from the Chest-Xray-Pneumonia and Qata-cov19 (Control Group II) datasets.

To train the three cascade systems, namely PCN_BV, PN_CBV, and PN_PC_BV. The first two-stage cascade system, PCN_BV, requires two sub-datasets, namely PCN_BV_D1 and PCN_BV_D2, where PCN_BV_D2 contains sub-categories (viral and bacterial) of the Non-Covid-19 Pneumonia class of PCN_BV_D1. Similarly, the second system PN_CBV also uses two sub-datasets, namely PN_CBV_D1 and PN_CBV_D2. PN_CBV_D2 is a subset of PN_CBV_D1 and contains the pneumonia sub-classes (COVID-19, bacterial, and viral). Beside, the three-stage cascade system PN_PC_BV requires three sub-datasets: PN_CBV_D1, PN_PC_BV_D2, and PCN_BV_D2. PN_CBV_D1 and PCN_BV_D2 are also present in the first two cascade systems, while PN_PC_BV_D2 is a new sub-dataset that differentiates COVID-19 from other non-COVID-19 pneumonias.

Table 1: The compiled COVID-QU-Ex_4C dataset

Source	Class	Number	Total
COVID-QU-Ex	Covid-19	11 956	31 175
	Normal	10 701	
Chest-Xray-Pneumonia + Qata-cov19 (Control Group 2)	Bacterial	5540	
	Viral	2978	

4.2 Results of the two-cascade strategies

Table 2 presents the results obtained from the two-stage cascade methods: PCN_BV and PN_CBV. It highlights all possible combinations between the CNN models. At each level, we fixed the first model in the first stage and varied the second model in the second stage. For instance, VGG16_VGG16 indicates that the VGG16 network was used in both levels, while VGG16_VGG16t indicates that the VGG16t model was obtained through transfer learning from the used VGG16 model in the first stage.

The results indicate that the models were less efficient when used independently compared to their combination with other second-stage models. These findings highlight the advantages of using cascade strategies, except for a few cases (22 out of 84). Surprisingly, when combined with other second-stage models using the PN_CBV method, all results decreased for Xception, which we believe is due to its low accuracy (96.31%) on the PN_CBV_D1 dataset compared to the other architectures.

The comparative study between the two-stage cascade methods PCN_BV and PN_CBV demonstrated the efficiency of PCN_BV for Xception, VGG16, InceptionV3, and DenseNet121. Specifically, accurate results were obtained for VGG16, InceptionV3, and DenseNet121 using PCN_BV, except when combined with DenseNet121 in all cases and with DenseNet201 for DenseNet121 and InceptionV3. In constract, PN_CBV was more accurate for VGG19 and DenseNet201, except when combining VGG19 with VGG16 or VGG19t and DenseNet201 with VGG16.

4.3 Results of the three-cascade strategy

For the two-stage cascade strategies, we generated 42 combinations for each method. However, for the three-stage cascade method, a high number of combinations can be

[1] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[2] https://github.com/lindawangg/COVID-Net/blob/master/docs/COVIDx.md

[3] https://www.kaggle.com/datasets/anasmohammedtahir/covidqu

[4] https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

[5] https://www.kaggle.com/datasets/aysendegerli/qatacov19-dataset

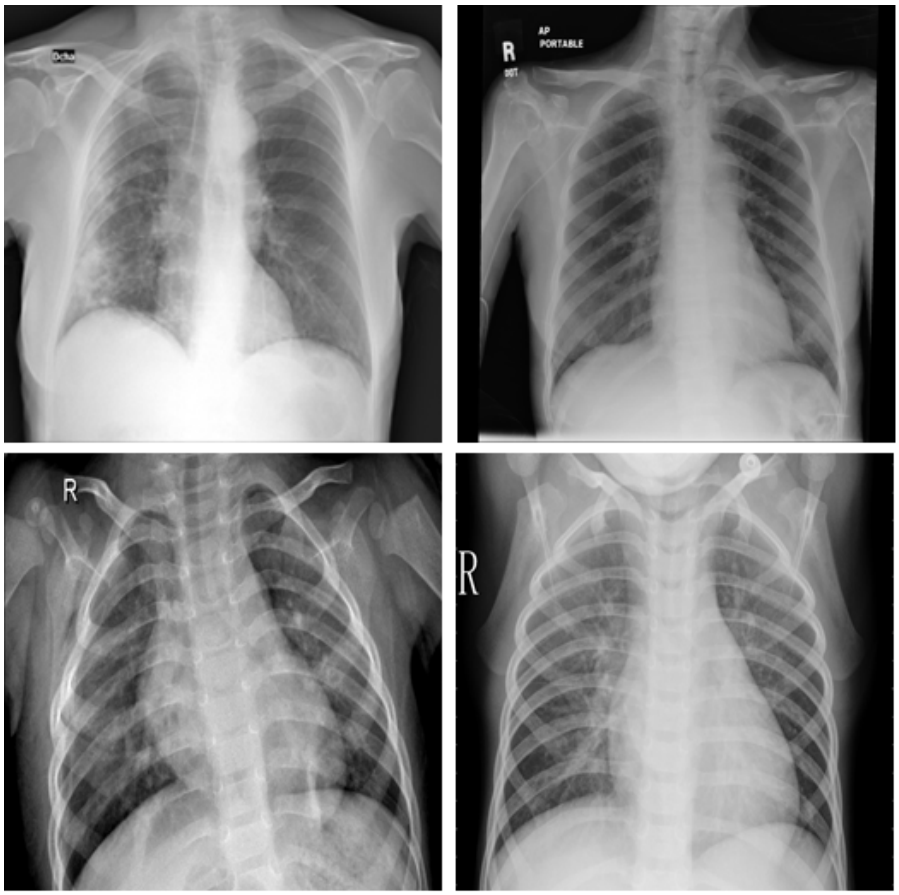


Figure 2: Samples from the COVID-QU-Ex_4C dataset

generated between the CNN architectures (294), which can be computationally expensive. Therefore, to reduce the number of combinations, we employed two strategies to select the appropriate model at each level. The first strategy combines CNNs based on transfer learning from the previous stage in a cascade strategy, where a unique architecture is used for each combination. The second strategy selects the two best models on the appropriate dataset at each level. For the first stage, we selected DenseNet121 and DenseNet201 due to their high accuracy on the PN_CBV_D1 dataset. In the second stage, we chose DenseNet121 and Xception, and in the third stage, we selected DenseNet201 and InceptionV3. Table 3 presents the obtained results based on the two strategies. In the first technique, DenseNet201 demonstrated its efficiency compared to other combinations. However, the results obtained by Xception were less promising. Besides, for the second strategy, the comparative study, shows that using Densenet201 in the first stage is more promising than DenseNet121. Additionally, in the second stage, DenseNet121 is more efficient than Xception. Overall, the ensemble's efficiency depends on the models' performance at the first and second stages. In general, the best models ensure the best combination.

4.4 Comparison and discussion

In the previous section, we conducted a comprehensive comparative study between obtained results on the COVID-QU-Ex_4C. Table 4 highlights the best strategy for each dataset, and overall, the DenseNets architectures yielded the best results. The obtained results demonstrate the challenges of distinguishing between pneumonia (92.31%) in both binary (bacterial and viral) and multi-class (COVID-19, bacterial, and viral) systems (96.80%). Whereas, CNNs accurately classified almost all images for datasets that classify pneumonia and COVID-19 (99.88%) or pneumonia and normal images (98.22%). These results confirm the advantages of using a cascade system to separate the four-class classification task based on annotation complexity, starting from the easiest to the most complex.

The comparative analysis of two-stage cascade systems indicates that PCN_BV is more suitable for Xception, VGG16, InceptionV3, and DenseNet121 models in the first stage, where it was less accurate in 16 among 42 cases. On the other hand, PN_CBV was more accurate for both VGG19 and DenseNet201. This method achieved the best result by DenseNet201_DenseNet121 architecture. Overall, these findings suggest that we cannot assume that PCN_BV presents the best strategy, as each technique has

Table 2: The obtained results on the two-stage cascade methods (PCN_BV, PN_CBV) in terms of accuracy (Acc), precision (P), recall (R), and F-measure (F1)

First Network	Hybridization	PCN_BV (%)				PN_CBV (%)			
		Acc	P	R	F1	Acc	P	R	F1
VGG16	VGG16	93.43	89.01	90.74	89.80	-	-	-	-
	VGG16_VGG16	95.56	93.81	92.68	93.16	94.77	93.05	91.09	91.80
	VGG16_VGG16t	**95.82**	**93.93**	**93.60**	**93.76**	95.14	93.12	92.57	92.82
	VGG16_VGG19	95.13	92.72	91.95	92.29	94.87	92.19	92.74	92.45
	VGG16_InceptionV3	95.72	94.11	93	93.48	95.35	93.34	92.89	93.08
	VGG16_Xception	95.61	93.64	93	93.29	95.27	92.81	93.13	92.97
	VGG16_DenseNet121	95.37	93.10	92.47	92.76	**95.80**	**93.69**	**94.47**	**94.06**
	VGG16_DenseNet201	95.77	93.92	93.44	93.66	95.70	94.25	93.55	93.85
VGG19	VGG19	94.21	90.50	92.48	91.26	-	-	-	-
	VGG19_VGG19	94.36	91.18	91.69	91.41	94.66	91.82	92.71	92.23
	VGG19_VGG19t	94.95	92.23	92.95	92.57	94.87	92.38	92.59	92.48
	VGG19_VGG16	94.87	92.49	92.62	92.51	94.53	92.67	90.94	91.54
	VGG19_InceptionV3	95	92.50	92.86	92.65	95.11	92.86	92.77	92.78
	VGG19_Xception	94.92	92.20	92.91	92.54	95.05	92.49	93.03	92.75
	VGG19_DenseNet121	94.68	91.78	92.41	92.08	**95.53**	**93.03**	**94.30**	**93.61**
	VGG19_DenseNet201	**95.05**	**92.40**	**93.26**	**92.82**	95.46	93.67	93.44	93.52
InceptionV3	InceptionV3	94.87	92.80	93.01	92.90	-	-	-	-
	InceptionV3_InceptionV3	95	93.24	92.85	93	94.69	92.49	92.56	92.50
	InceptionV3_InceptionV3t	**95.11**	**93.13**	**93.46**	**93.29**	95	92.82	93.17	92.97
	InceptionV3_VGG16	94.79	92.96	92.41	92.62	94.10	92.29	90.69	91.23
	InceptionV3_VGG19	94.33	91.75	91.60	91.64	94.28	91.54	92.52	91.99
	InceptionV3_Xception	94.94	92.90	92.94	92.90	94.63	92.07	92.80	92.43
	InceptionV3_DenseNet121	94.68	92.38	92.40	92.37	**95.10**	**92.70**	**94.07**	**93.33**
	InceptionV3_DenseNet201	95.02	92.98	93.15	93.05	95.03	93.32	93.19	93.22
Xception	Xception	95	93.17	92.92	93	-	-	-	-
	Xception_Xception	95.21	92.73	93.24	92.97	93.78	90.54	92.26	91.31
	Xception_Xceptiont	**95.27**	**92.89**	**93.29**	**93.07**	93.88	90.13	92.14	91.01
	Xception_VGG16	95.06	92.79	92.70	92.69	93.27	90.84	90.16	90.22
	Xception_ VGG19	94.60	91.62	91.89	91.73	93.36	89.80	91.87	90.69
	Xception_InceptionV3	95.22	92.86	93.03	92.91	93.85	90.77	92.01	91.31
	Xception_DenseNet121	94.94	92.22	92.66	92.43	**94.26**	**90.91**	**93.53**	**92**
	Xception_DenseNet201	95.24	92.77	93.34	93.04	94.21	91.39	92.69	91.96
DenseNet121	DenseNet121	95.43	92.56	93.05	92.80	-	-	-	-
	DenseNet121_DenseNet121	95.59	93.03	93.14	93.07	**96.11**	**93.76**	**94.74**	**94.22**
	DenseNet121_DenseNet121t	**95.99**	**93.47**	**94.27**	**93.85**	95.56	93.18	93.04	93.07
	DenseNet121_VGG16	95.74	93.62	93.23	93.37	95.05	93.18	91.33	91.98
	DenseNet121_VGG19	95.21	92.33	92.27	92.27	95.16	92.28	93.06	92.64
	DenseNet121_InceptionV3	95.86	93.70	93.46	93.54	95.66	93.33	93.23	93.25
	DenseNet121_Xception	95.80	93.47	93.57	93.51	95.62	93.05	93.56	93.30
	DenseNet121_DenseNet201	95.88	93.56	93.82	93.68	96.04	94.34	93.93	94.09
DenseNet201	DenseNet201	95.45	92.75	93.71	93.20	-	-	-	-
	DenseNet201_DenseNet201	**95.75**	**93.50**	**93.69**	**93.59**	96.22	94.57	93.97	94.21
	DenseNet201_DenseNet201t	95.64	93.09	93.40	93.24	95.98	93.60	93.85	93.72
	DenseNet201_VGG16	95.61	93.55	93.10	93.27	95.26	93.50	91.45	92.18
	DenseNet201_VGG19	95.11	92.29	92.21	92.22	95.42	92.65	93.26	92.93
	DenseNet201_InceptionV3	95.75	93.69	93.38	93.49	95.83	93.57	93.28	93.39
	DenseNet201_Xception	95.69	93.38	93.51	93.43	95.75	93.18	93.49	93.33
	DenseNet201_DenseNet121	95.48	92.94	93.06	92.99	**96.27**	**93.84**	**94.79**	**94.28**

its advantages depending on the specified architecture.

Figures 3 and 4 compare the results obtained from transfer learning from ImageNet and transfer learning within the three cascade systems: PCN_BV, PN_CBV, and PN_PC_BV. In general, transfer learning within the cascade systems based on PCN_BV was more promising, except for DenseNets, where transfer learning from ImageNet based on PN_CBV yielded accurate results. The comparative study between the used methods for transfer learning within the cascade systems also highlights the efficiency

Table 3: The obtained results based on the three-stage cascade strategy PN_PC_BV in terms of accuracy (Acc), precision (P), recall (R), and F-measure (F1)

Network	Acc (%)	P (%)	R (%)	F1 (%)
VGG16_VGG16t_VGG16t	95.21	92.97	92.74	92.84
VGG19_VGG19t_VGG19t	94.76	92.22	92.14	92.15
InceptionV3_InceptionV3t_InceptionV3t	94.77	92.74	92.70	92.68
Xception_Xceptiont_Xceptiont	93.75	90.94	91.62	91.16
DenseNet121_DenseNet121t_DenseNet121t	**95.62**	**92.90**	**93.71**	**93.29**
DenseNet201_DenseNet201t_DenseNet201t	**96.12**	**93.86**	**94.08**	**93.96**
Densenet201_ Densenet121_DenseNet201	**95.99**	**93.63**	**93.83**	**93.72**
Densenet201_ Xception_DenseNet201	95.96	93.64	93.77	93.70
Densenet201_ Densenet121_InceptionV3	**95.98**	**93.79**	**93.47**	**93.59**
Densenet201_ Xception_InceptionV3	95.94	93.80	93.42	93.56
Densenet121_ Densenet121_DenseNet201	95.78	93.49	93.71	93.59
Densenet121_ Xception_DenseNet201	95.75	93.47	93.65	93.55
Densenet121_ Densenet121_InceptionV3	95.80	93.67	93.42	93.51
Densenet121_ Xception_InceptionV3	95.77	93.64	93.36	93.46

Table 4: The best results for each dataset

Dataset	Method	Accuracy(%)
COVID-QU-Ex_4C	DenseNet201	95.45
PCN_BV_D1	DenseNet121	98.11
PCN_BV_D2	VGG19, InceptionV3	92.31
PN_CBV_D1	DenseNet201	98.22
PN_CBV_D2	DenseNet201	96.80
PN_PC_BV_D2	VGG16	99.88

of PCN_BV over both PN_CBV and PN_PC_BV. In conclusion, the transfer learning strategy within the two-stage cascade system was not suitable for DenseNets and was less efficient for the three-stage cascade systems.

Figure 5 presents a comparative analysis of the best combinations for each network and strategy. For the PCN_BV strategy, the most accurate results were obtained by merging models with their transferred versions and employing DenseNet201 in the second stage. The transfer learning within the cascade systems was observed to be more promising for the PCN_BV strategy. Conversely, for PN_CBV, combining models with DenseNet121 was more accurate. Based on these findings, we propose that using transfer learning from multi-class classification problems in two-stage cascade systems is more promising.

Table 5 displays the best results for each strategy, highlighting the efficiency of two-stage cascade systems compared to the three-stage cascade system. It's worth noting that while PCN_BV and PN_PC_BV achieved the same result, PCN_BV is more storage-efficient as it only requires storing two models instead of three for prediction. According to the best results of two-stage cascade systems, we can conclude that the performance of the first-stage models influences the accuracy of the two-stage cascade strategies. For PCN_BV, DenseNet121_DenseNet121t achieved the best result due to the high performance of DenseNet121 in the first stage on the PCN_BV_D1 dataset. Similarly, for PN_CBV, the interesting results of DenseNet201 in the first stage generated the best combination (DenseNet201_DenseNet121).

Table 5: The best results of the cascade strategies

Strategy	Accuracy (%)
PCN_BV	95.99
PN_CBV	96.27
PN_PC_BV	95.99

Table 6 presents the average computational complexity of each model when combined with the remaining models in the second stage. It compares the computational complexity of the non-cascade system with the average computational complexity of the PCN_BV and PN_CBV strategies. Overall, the comparative study highlights that in the non-cascade strategy, InceptionV3 and Xception are more computationally expensive compared to the other models, with DenseNet121 being the fastest, with a training time of 40 minutes. On the other hand, the study shows that the PCN_BV and PN_CBV strategies are more computationally expensive compared to the non-cascade system, except for InceptionV3. In general, the training time of PCN_BV is slightly closer to that of the non-cascade system. However, the PN_CBV strategy took longer compared to both the PCN_BV strategy and the non-cascade system.

Table 7 highlights the training time of the PN_PC_BV strategy, showing its high complexity compared to the two cascade strategies, except in a few cases where the Xception_Xceptiont_Xceptiont and VGG19_VGG19t_VGG19t strategies required 2 hours for training.

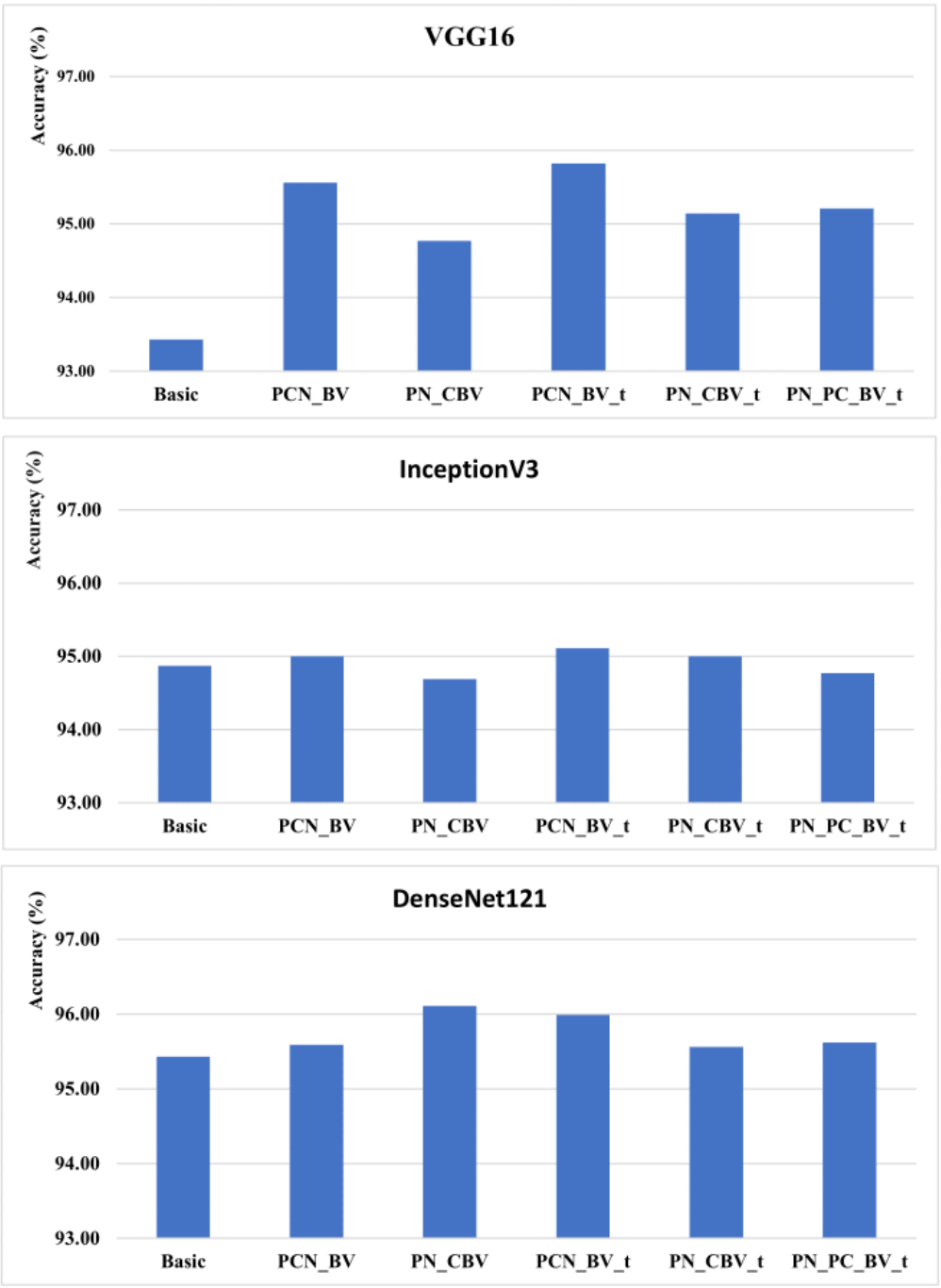


Figure 3: Comparison between transfer learning from ImageNet and transfer learning within the three cascade systems (1)

5 Conclusion

In this study, we proposed three cascade systems based on six CNN architectures for COVID-19 classification: two types of two-stage cascade strategies (PCN_BV and PN_CBV) and a three-stage cascade system (PN_PC_BV). To improve results, we integrated transfer learning strategies within the proposed systems. We validated the proposed methods on a newly generated dataset (COVID-QU-Ex_4C) that contains four classes: COVID-19, normal, bacterial, and viral. The comparative study showed that the two-stage cascade systems were more efficient than the three-stage cascade system. Furthermore, the results of the two-stage cascade systems revealed that PCN_BV was more accurate and faster in training in most cases, and transfer learning within the cascade systems was more effective

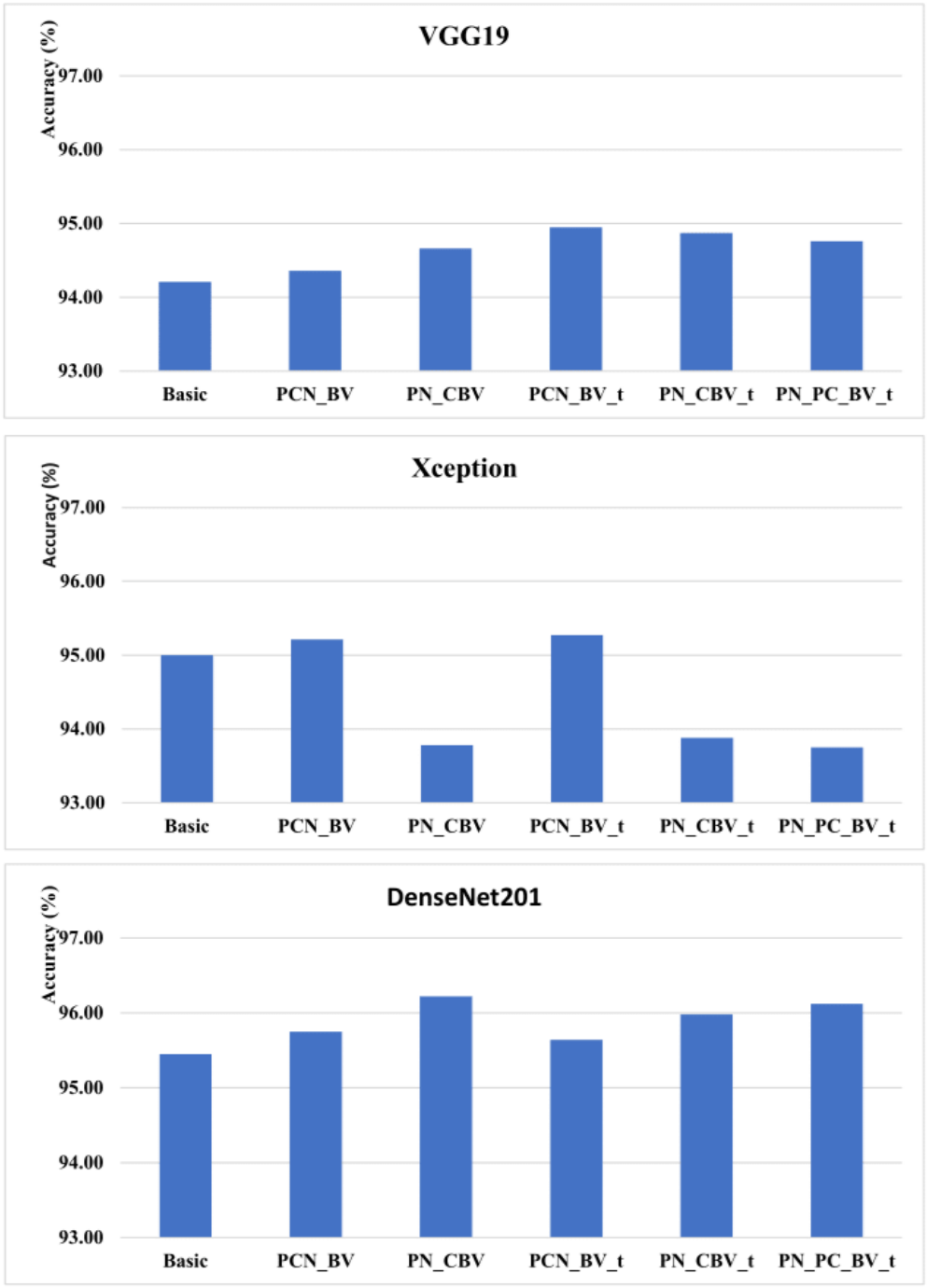


Figure 4: Comparison between transfer learning from ImageNet and transfer learning within the three cascade systems (2)

for PCN_BV than PN_CBV.

As a future perspective, we plan to introduce more variability into the proposed dataset and address the data imbalance issue. We propose to explore additional data augmentation techniques, such as generative adversarial networks (GANs), to tackle this problem. Additionally, we suggest introducing Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize activated features during detection.

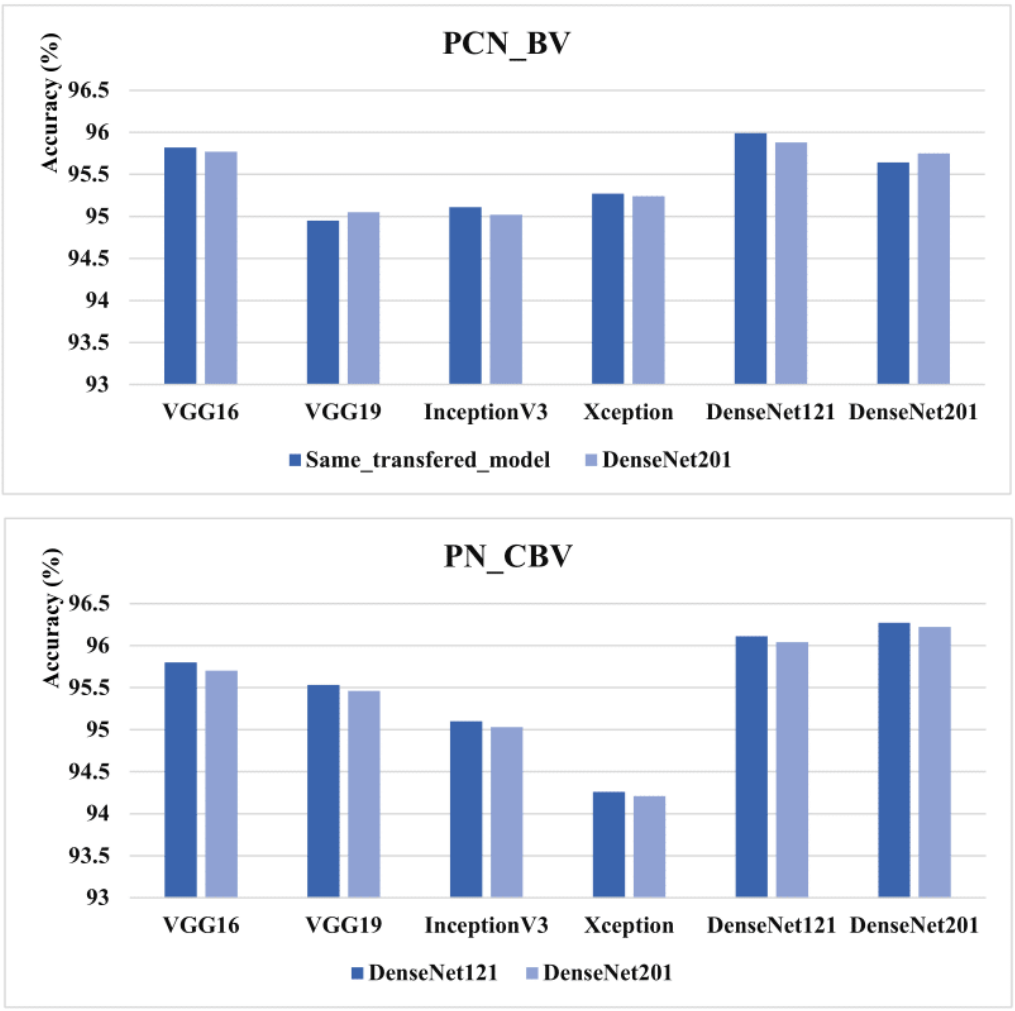


Figure 5: The two best combinations for each network and strategy

Table 6: The average training time of the two-stage cascade strategies (Hours:Minutes:Seconds).

First Network	Non cascade	PCN_BV	PN_CBV
VGG16	00:59:45	01:08:04	01:34:13
VGG19	01:08:04	01:16:14	01:40:31
InceptionV3	01:32:18	01:05:51	01:18:36
Xception	01:11:55	01:20:26	01:44:37
DenseNet121	00:40:47	00:50:36	01:09:58
DenseNet201	00:56:35	01:11:02	01:30:22

Table 7: Training time of the three-stage cascade strategy PN_PC_BV (Hours:Minutes:Seconds)

Network	PN_PC_BV
VGG16_VGG16t_VGG16t	01:53:37
VGG19_VGG19t_VGG19t	02:02:49
InceptionV3_InceptionV3t_InceptionV3t	01:16:46
Xception_Xceptiont_Xceptiont	02:10:01
DenseNet121_DenseNet121t_DenseNet121t	01:11:15
DenseNet201_DenseNet201t_DenseNet201t	01:48:39
Densenet201_Densenet121_DenseNet201	01:36:02
Densenet201_Xception_DenseNet201	01:54:36
Densenet201_Densenet121_InceptionV3	01:37:50
Densenet201_Xception_InceptionV3	01:56:24
Densenet121_Densenet121_DenseNet201	01:16:50
Densenet121_Xception_DenseNet201	01:35:24
Densenet121_Densenet121_InceptionV3	01:18:38
Densenet121_Xception_InceptionV3	01:37:12

References

[1] D. Cucinotta and M. Vanelli, "Who declares covid-19 a pandemic," *Acta Bio Medica: Atenei Parmensis*, vol. 91, no. 1, p. 157, 2020, doi: https://doi.org/10.23750/abm.v91i1.9397.

[2] F. Yu, L. Du, D. M. Ojcius, C. Pan, and S. Jiang, "Measures for diagnosing and treating infections by a novel coronavirus responsible for a pneumonia outbreak originating in wuhan, china," *Microbes and infection*, vol. 22, no. 2, pp. 74–79, 2020, doi: https://doi.org/10.1016/j.micinf.2020.01.003.

[3] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, "Correlation of chest ct and rt-pcr testing in coronavirus disease 2019 (covid-19) in china: a report of 1014 cases," *Radiology*, 2020, doi: https://doi.org/10.1148/radiol.2020200642.

[4] H. L. Fred, "Drawbacks and limitations of computed tomography: views from a medical educator," *Texas Heart Institute Journal*, vol. 31, no. 4, p. 345, 2004.

[5] L. Brunese, F. Martinelli, F. Mercaldo, and A. Santone, "Machine learning for coronavirus covid-19 detection from chest x-rays," *Procedia Computer Science*, vol. 176, pp. 2212–2221, 2020, doi: https://doi.org/10.1016/j.procs.2020.09.258.

[6] S. Bharati, P. Podder, M. Mondal, and V. Prasath, "Medical imaging with deep learning for covid-19 diagnosis: a comprehensive review," *arXiv preprint arXiv:2107.09602*, 2021, doi: https://doi.org/10.48550/arXiv.2107.09602.

[7] R. Sadoon and A. Chaid, "Classification of pulmonary diseases using a deep learning stacking ensemble model," *Informatica*, vol. 48, no. 14, 2024, doi: https://doi.org/10.31449/inf.v48i14.6145.

[8] Z. A. Oraibi and S. Albasri, "A robust end-to-end cnn architecture for efficient covid-19 prediction form x-ray images with imbalanced data," *Informatica*, vol. 47, no. 7, 2023, doi: https://doi.org/10.31449/inf.v47i7.4790.

[9] V. Gupta, N. Jain, J. Sachdeva, M. Gupta, S. Mohan, M. Y. Bajuri, and A. Ahmadian, "Improved covid-19 detection with chest x-ray images using deep learning," *Multimedia Tools and Applications*, vol. 81, no. 26, pp. 37 657–37 680, 2022, doi: https://doi.org/10.1007/s11042-022-13509-4.

[10] E. Jangam, C. S. R. Annavarapu, and A. A. D. Barreto, "A multi-class classification framework for disease screening and disease diagnosis of covid-19 from chest x-ray images," *Multimedia Tools and Applications*, pp. 1–35, 2022, doi: https://doi.org/10.1007/s11042-022-13710-5.

[11] P. Sharma, R. Arya, R. Verma, and B. Verma, "Convcapsnet: capsule based network for covid-19 detection through x-ray scans," *Multimedia Tools and Applications*, pp. 1–25, 2023, doi: https://doi.org/10.1007/s11042-023-14353-w.

[12] H. Kör, H. Erbay, and A. H. Yurttakal, "Diagnosing and differentiating viral pneumonia and covid-19 using x-ray images," *Multimedia Tools and Applications*, vol. 81, no. 27, pp. 39 041–39 057, 2022, doi: https://doi.org/10.1007/s11042-022-13071-z.

[13] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1207–1220, 2021, doi: https://doi.org/10.1007/s10044-021-00984-y.

[14] M. E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. Al Emadi *et al.*, "Can ai help in screening viral and covid-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020, doi: https://doi.org/10.1109/access.2020.3010287.

[15] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan *et al.*, "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in biology and medicine*, vol. 132, p. 104319, 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104319.

[16] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, "Deep learning based detection and analysis of covid-19 on chest x-ray images," *Applied Intelligence*, vol. 51, no. 3, pp. 1690–1700, 2021, doi: https://doi.org/10.1007/s10489-020-01902-1.

[17] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng *et al.*, "A deep learning algorithm using ct images to screen for corona virus disease (covid-19)," *European radiology*, vol. 31, no. 8, pp. 6096–6104, 2021, doi: https://doi.org/10.1007/s00330-021-07715-1.

[18] C. Ouchicha, O. Ammor, and M. Meknassi, "Cvdnet: A novel deep learning architecture for detection of coronavirus (covid-19) from chest x-ray images," *Chaos, Solitons & Fractals*, vol. 140, p. 110245, 2020, doi: https://doi.org/10.1016/j.chaos.2020.110245.

[19] A. M. Ismael and A. Şengür, "Deep learning approaches for covid-19 detection based on chest x-ray images," *Expert Systems with Applications*, vol. 164, p. 114054, 2021, doi: https://doi.org/10.1109/access.2020.2989273.

[20] H. Gunraj, A. Sabri, D. Koff, and A. Wong, "Covid-net ct-2: Enhanced deep neural networks for detection of covid-19 from chest ct images through bigger, more diverse learning," *Frontiers in Medicine*, vol. 8, p. 3126, 2022, doi: https://doi.org/10.3389/fmed.2021.729287.

[21] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, and R. B. Pachori, "Application of deep learning techniques for detection of covid-19 cases using chest x-ray images: A comprehensive study," *Biomedical Signal Processing and Control*, vol. 64, p. 102365, 2021, doi: https://doi.org/10.1016/j.bspc.2020.102365.

[22] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images," *Informatics in medicine unlocked*, vol. 20, p. 100412, 2020, doi: https://doi.org/10.1016/j.imu.2020.100412.

[23] C. Öksüz, O. Urhan, and M. K. Güllü, "Ensemble-cvdnet: A deep learning based end-to-end classification framework for covid-19 detection using ensembles of networks," *arXiv preprint arXiv:2012.09132*, 2020, doi: https://doi.org/10.48550/arXiv.2012.09132.

[24] K. Y. Win, N. Maneerat, S. Sreng, and K. Hamamoto, "Ensemble deep learning for the detection of covid-19 in unbalanced chest x-ray dataset," *Applied Sciences*, vol. 11, no. 22, p. 10528, 2021, doi: https://doi.org/10.3390/app112210528.

[25] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, "Explainable deep learning for pulmonary disease and coronavirus covid-19 detection from x-rays," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105608, 2020, doi: https://doi.org/10.1016/j.cmpb.2020.105608.

[26] A. Shelke, M. Inamdar, V. Shah, A. Tiwari, A. Hussain, T. Chafekar, and N. Mehendale, "Chest x-ray classification using deep learning for automated covid-19 screening," *SN computer science*, vol. 2, no. 4, pp. 1–9, 2021, doi: https://doi.org/10.1007/s42979-021-00695-5.

[27] A. M. Tahir, M. E. Chowdhury, A. Khandakar, T. Rahman, Y. Qiblawey, U. Khurshid, S. Kiranyaz, N. Ibtehaz, M. S. Rahman, S. Al-Maadeed *et al.*, "Covid-19 infection localization and severity grading from chest x-ray images," *Computers in biology and medicine*, vol. 139, p. 105002, 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.105002.

[28] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018, doi: https://doi.org/10.1016/j.cell.2018.02.010.

Explainable AI for Pancreatic Cancer Prediction and Survival Prognosis: An Interpretable Deep Learning and Machine Learning Approach

Srinidhi B, M S Bhargavi*
Department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, India
E-mail: srinidhib18@gmail.com, ms.bhargavi@gmail.com
*Corresponding Author

Keywords: explainable artificial intelligence, shapley additive explanations (SHAP), local interpretable model-agnostic explanations (LIME), deep learning, pancreatic cancer

Received: September 2, 2023

Pancreatic cancer's devastating impact and low survival rates call for improved detection methods. While Artificial Intelligence has shown remarkable progress, its increasing complexity has led to "black box" models, hindering their acceptance in critical fields like healthcare. To address this, Explainable Artificial Intelligence (XAI) has gained traction, aiming to create transparent AI systems. In this study, we propose a comprehensive approach that combines the power of Deep Learning for pancreatic cancer detection using Computed Tomography (CT) images and Machine Learning (ML) for survival prognosis based on clinical data. By leveraging CT images with Deep learning models such as Convolutional Neural Networks, VGG-16 and DenseNet-201, effective diagnosis of Pancreatic Cancer is achieved and comprehensive insights into the tumor's spatial characteristics are obtained. DenseNet-201 outperformed the other models in terms of accuracy and interpretability with a predictive accuracy of 95%. The integration of ML techniques such as Stochastic Gradient Descent, Naïve Bayes and Extra Tree classifiers with clinical data predicts the chances of survival, providing vital information for treatment planning and personalized care. To validate the model's accuracy and interpretability, a comprehensive XAI validation is conducted using state-of-the-art techniques like Local Interpretable Model-agnostic Explanations and Shapley Additive Explanations. These methods provide localized explanations for predictions, allowing clinicians to understand risk and survival chances. This study holds immense potential to aid healthcare professionals in diagnosis, prognosis, and personalized treatment strategies, contributing to enhanced patient outcomes in the fight against pancreatic cancer.

Povzetek: Analizirana sta napovedovanje in prognoza preživetja pri raku trebušne slinavke z uporabo globokega učenja in razložljive umetne inteligence (XAI) za interpretacijo napovedi, izboljšanje zaupanja v AI.

1 Introduction

Pancreatic cancer is a devastating malignancy that poses a significant global health challenge due to its aggressive nature and poor prognosis [1]. Despite advances in medical research and cancer treatments, the five-year survival rate for pancreatic cancer patients remains dishearteningly low, with many cases being diagnosed at advanced stages when effective interventions become limited. Cancer detection and chance of survival are critical factors that can significantly impact patient outcomes by enabling timely treatments and personalized therapeutic strategies. Consequently, there is an urgent need to explore novel approaches that offer deeper insights into pancreatic cancer development and patient survival rates.

Pancreatic cancer's clinical landscape is characterized by its insidious nature in the development stages, often presenting vague or non-specific symptoms. This lack of distinct clinical indicators makes early diagnosis challenging, resulting in delayed treatment initiation and reduced chances of successful interventions. Additionally, the intricate biology of pancreatic cancer necessitates a comprehensive understanding of the underlying molecular and genetic factors driving disease progression and influencing patient outcomes [2]. As such, the development of accurate predictive models that can identify individuals at high risk of developing pancreatic cancer and estimate patient survival probabilities becomes imperative in the fight against this lethal disease.

While current medical technologies have significantly contributed to cancer detection and diagnosis, they still face limitations. Imaging modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and endoscopic ultrasound provide valuable visualizations of pancreatic tumors and staging information. However, these imaging techniques may miss small lesions or produce ambiguous results in anatomically complex areas, leading to diagnostic

challenges. Furthermore, the use of biomarker analysis, including CA 19-9 and several other factors for monitoring treatment response and assessing disease progression may lack specificity and sensitivity, resulting in false-positive or false-negative outcomes. These limitations highlight the need for more accurate and comprehensive predictive tools for improved cancer detection and patient survival estimation.

Advancements in AI technology enabled the rapid and precise identification of cancer through diverse medical imaging methods. The Deep Learning (DL) algorithms have garnered significant attention in medical imaging analysis. DL algorithms, particularly Convolutional Neural Networks (CNNs), can automatically learn hierarchical representations from raw image data, such as CT scans. This unique capability enables DL models to detect complex patterns and features in medical images, offering great potential for aiding the detection and diagnosis of pancreatic cancer [3]. Additionally, deep learning can be integrated with clinical data, including patient demographics and medical history, to develop comprehensive predictive models that deliver more accurate and personalized survival estimates

Machine learning (ML) models can be trained on diverse clinical data to predict patient survival rates by considering factors such as age, tumor grade, CA 19-9 levels, sex, and medical history. By leveraging these ML-based approaches, clinicians can obtain individualized survival chances, which are essential for tailoring treatment plans and optimizing patient care. However, DL and conventional ML models often operate as black boxes, lacking transparency in their decision-making process. This opacity hinders the widespread adoption of DL and ML in critical clinical decision-making settings, where understanding the factors influencing predictions is paramount.

To address the interpretability challenge associated with DL and ML models, XAI techniques have emerged as a solution [4]. In this study, we propose a comprehensive approach that combines the power of Deep Learning for pancreatic cancer detection using CT images and Machine Learning for survival prognosis based on clinical data. To enhance the interpretability and trustworthiness of our model, we incorporate XAI techniques, specifically Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP).

LIME plays a crucial role in providing interpretable insights into the DL-based cancer detection model's predictions. By employing LIME, we can identify the specific regions within the CT images that significantly contribute to the model's decision-making process. These regions act as crucial markers for the presence of tumors and other abnormalities, enabling radiologists and clinicians to validate and understand the model's findings. Mapping CT image regions relevant to the model's predictions not only boosts confidence in AI-driven detection but also assists medical professionals in recognizing subtle signs of pancreatic cancer, potentially enabling earlier diagnosis and timely intervention.

Additionally, SHAP plays a pivotal role in explaining machine learning-based survival rate predictions using clinical data. SHAP provides a quantitative measure of the influence of each clinical feature on the model's survival rate estimations. By identifying the most affected features, SHAP empowers healthcare professionals to understand the factors driving the model's predictions and their relative importance in determining patient survival probabilities. This level of interpretability allows clinicians to prioritize critical clinical factors and consider them in treatment planning and patient management. SHAP's ability to reveal the most influential clinical factors further enhances the personalized nature of our model's survival predictions, ensuring tailored and optimized patient care.

By integrating LIME and SHAP into our interpretable analytical model, we provide a comprehensive framework that not only accurately detects pancreatic cancer using CT images but also offers transparent and interpretable survival rate predictions based on clinical data. The combination of these XAI techniques not only bolsters the trustworthiness of the model's predictions but also empowers medical professionals to make informed decisions based on AI-driven insights. Ultimately, our model's enhanced interpretability contributes to improved patient outcomes by facilitating earlier detection, more precise prognostication, and personalized treatment strategies for individuals battling pancreatic cancer.

The remainder of the paper is organized as follows. Section I provides the Introduction to the research. Section II presents the literature review of existing systems. Section III provides Dataset description. Section IV presents the proposed methodology and implementation details. Section V presents the results and discussion of the proposed system. Section VI concludes the work with future scope.

2 Related work

Pancreatic cancer remains a formidable and deadly malignancy, necessitating innovative approaches to improve early detection and prognostication for enhanced patient outcomes. The recent convergence of DL for cancer detection using CT images, ML for survival rate prediction based on clinical data, and the integration of XAI techniques have shown great potential in tackling the challenges presented by this devastating disease.

The incorporation of XAI methods, such as LIME and SHAP, enhances the interpretability and trustworthiness of AI-driven models, empowering medical professionals to comprehend the factors driving predictions and make well-informed clinical decisions. In this literature survey, we delve into research focused on the development and application of interpretable analytical models for pancreatic cancer detection and survival prognosis. By examining the advances and insights from these studies, we aim to provide a comprehensive understanding of the state-of-the-art techniques that hold the potential to revolutionize

pancreatic cancer management and contribute to improved patient care.

The recent convergence of DL techniques for cancer detection using CT images has shown significant promise in addressing the challenges posed by this devastating disease. Among the various DL approaches, CNNs and their extensions, such as recurrent CNNs, have emerged as particularly effective in the detection of pancreatic cancer. These methods have demonstrated impressive accuracy in identifying pancreatic tumors and differentiating them from normal tissues, as evidenced by multiple studies [3,5,6,7,8,9,10,11]. CNNs are well-suited for processing the complex and high-dimensional data found in medical imaging, enabling them to extract intricate patterns and features that are indicative of cancerous tissues. The effectiveness of these models is largely attributed to their ability to learn from large datasets, which enhances their capacity to recognize subtle distinctions between malignant and benign tissues. Recurrent CNNs, which integrate temporal and spatial information, have further advanced the field by improving the detection accuracy and robustness of these models. This approach allows for the consideration of sequential dependencies in image slices, which is crucial for the accurate identification of tumors in CT scans. The application of DL models offers significant advantages over traditional diagnostic methods, including faster processing of large volumes of image data and higher accuracy in tumor detection. As a result, these advancements hold the potential to facilitate earlier diagnosis, thereby improving treatment outcomes.

To address the interpretability challenge associated with DL and ML models, XAI techniques have emerged as a solution. The incorporation of XAI methods, such as LIME and SHAP enhances the interpretability and trustworthiness of AI-driven models, empowering medical professionals to comprehend the factors driving predictions and make well-informed clinical decisions. To better understand the model's decision-making process, a limited number of studies have explored the use of LIME and SHAP for predictions. LIME, in particular, offers valuable insights by mapping regions on CT images and highlighting specific areas that influence the model's predictions and an attempt for such analysis has been made in [12]. In some research, LIME has been utilized for feature importance analysis in predicting Pancreatic cancer [13,14,15].

Survival prediction in pancreatic cancer aids in improved treatment planning, provides realistic prognostic information for patients and their families, and supports the clinical trial design and assessment of new therapies. Research on predicting patient survival rates for pancreatic cancer using machine learning models based on clinical data has revealed promising results, demonstrating the potential to significantly improve patient outcomes. Various ML algorithms, such as random forests, support vector machines, logistic regression, nearest neighbor and gradient boosting, have been explored and assessed for their performance in survival prediction [16,17]. These models utilize extensive clinical datasets to identify patterns and correlations that might not be apparent to human clinicians. The effectiveness of these models in survival prediction is attributed to their capacity to process and analyze complex data, offering more accurate prognostic insights than traditional methods.

Despite these advancements, the lack of interpretability in these models poses a significant challenge, hindering their seamless integration into clinical practice and limiting clinician trust. To address this limitation XAI methods like SHAP have been incorporated [12,18,19]. The incorporation of SHAP was emphasized to elucidate the influential clinical factors driving survival predictions, aiding clinicians in making data-driven decisions for personalized treatment planning.

The literature review reveals that there have been limited research efforts focused on developing interpretable analytical models for pancreatic cancer detection and survival prognosis. The application of interpretability in this field is still in its early stages, with only a limited number of studies available. Table 1 provides a summary of related works that utilize explainable AI models for pancreatic cancer prediction and survival prognosis. The integration of XAI techniques in these studies emphasizes the significance of transparency and trustworthiness in AI-driven medical applications. As these innovative methodologies continue to evolve, they hold the potential to revolutionize pancreatic cancer management and improve patient outcomes.

Another notable limitation observed across the surveyed papers is the absence of utilization of multiple models of DL and ML for interpretation and validation. The studies predominantly focused on applying specific XAI techniques to individual models, such as using LIME or SHAP for specific DL or ML models. Consequently, there was a lack of exploration into the advantages of employing a diverse set of DL and ML models for these purposes. This oversight hinders comprehensive comparison and validation of XAI techniques across different model architectures. There is a requirement to consider adopting multiple, diverse DL and ML models in their XAI framework to gain a more holistic understanding of the predictive features and foster the standardization of XAI practices in pancreatic cancer detection and survival prediction.

In conclusion, the literature survey has shed light on the progress made in developing interpretable analytical models for pancreatic cancer detection and survival prediction. The combination of DL, ML, and XAI techniques holds significant potential for enhancing early detection, personalized treatment planning, and ultimately improving patient care.

Table 1: Summary of related works with explainability of models in pancreatic cancer prediction and survival prognosis.

Authors	Focus Area	Models	Dataset	Performance Metrics	XAI Technique	Key Findings
Goel et al. (2021) [15]	Pancreatic Cancer Detection	Logistic Regression, Adaboost, Neural Network, ensemble model	Gene Expression Omnibus-miRNA biomarkers	Sensitivity – 0.85,0.90,0.88, 0.95. Specificity – 0.98,0.94,0.96, 0.98.	LIME SHAP	Ensemble models with feature selection and improved diagnosis accuracy for pancreatic cancer detection
Srinidhi B et. al. (2023) [12]	Pancreatic Cancer Detection	CNN SVM	CT Images, Clinical Data from - The Cancer Imaging Archive	None	LIME SHAP	Combined XAI with CNN and SVM for model interpretability on prediction
Bobes-Bascarán et al. (2024) [13]	Pancreatic Cancer Detection	Decision Trees, Random Forest, XGBoost	Clinical Data from -The Cancer Genome Atlas Program	Accuracy - 0.66,0.54,0.66 Precision - 0.72,0.76,0.76 Recall - 0.66,0.54,0.66	LIME SHAP	Feature importance analysis for Pancreatic Cancer
Dimitris et al. (2021) [19]	Survival Prognosis	Optimal Classification Tree, XGBoost	Clinical data of 2,784 Patients	1-year AUC for OCT - 0.63 3-year AUC - 0.67	SHAP	SHAP was utilized to identify the best predictor
Keyl et al. (2022) [18]	Survival Prognosis	Random Survival Forest	Albumin, CT image, Radiomics data, Molecular data of 203 Cohort	C-index - 0.71	SHAP	Feature importance analysis for survival prognosis from multi-modal data

3 Dataset description

In this research, we utilized the Cancer Imaging Archive (TCIA) dataset [20], a comprehensive collection of medical images and clinical data, to develop an interpretable analytical model for pancreatic cancer detection and survival rate prediction. The dataset comprises 100 cancer-positive CT images and 100 cancer-negative CT images, offering a diverse set of cases for robust model training and evaluation. For cancer detection, we employed an 80-20% split, with 80% of the images used for training the model and 20% for testing its performance. Furthermore, for survival prognosis, we utilized seven critical clinical features, including age, gender, KI-67 Index, PFS Months, Creatine, CA 19/9 U/ml and tumor grade. Around 50 patient records were considered for this work to create a well-rounded prognostic model. Similar to the cancer detection model, the survival prediction model was trained on 80% of the clinical data and evaluated on the remaining 20% to ensure reliable and generalizable results.

4 Methodology

The proposed system in this research aims to develop an interpretable analytical model for pancreatic cancer detection and survival prognosis, leveraging a combination of CT images and clinical data. The preprocessing pipeline for the CT images involves initial RGB conversion, normalization, and conversion to an array format, ensuring standardized input for subsequent analysis. These preprocessed images are then fed to Deep Learning models such as CNN [21], VGG-16 [22] and

DenseNet-201 [23] to extract meaningful features and perform accurate cancer predictions. To enhance the interpretability of the deep learning model, the LIME algorithm is applied. LIME generates region marking for the CT images, highlighting specific areas influencing the model's predictions. This approach empowers radiologists and medical professionals to validate the model's decision process, instilling confidence in the AI-driven diagnostic tool and enabling more informed decision-making for early cancer detection.

In parallel, the clinical data is preprocessed through label encoding and the removal of irrelevant columns to ensure data quality and uniformity. This preprocessed clinical data is then utilized to train and evaluate different machine learning models, such as Stochastic Gradient Descent (SGD) [24], Extra Tree [25], and Naïve Bayes [26], for survival rate prediction. To ensure interpretability in the machine learning models' survival predictions, the SHAP algorithm is employed. SHAP provides insights into the most influential clinical features affecting the model's prognostic outcomes, empowering medical practitioners to make data-driven decisions and design personalized treatment plans.

The suggested system's architecture is depicted in Figure 1. At the initial level of the architecture, data processing occurs, providing input for the deep learning and machine learning models. The outcomes are then validated using explainable artificial intelligence techniques. For cancer prediction, the data preparation includes image conversion, normalization, and image-to-array. The pre-processed CT images will be fed to the DL models, and the predictions made by these models (CNN, VGG-16, DenseNet-201) will be validated using LIME. Similarly, the Machine learning models (Extra Tree, SGD, Naïve Bayes) are applied to the pre-processed clinical dataset. To validate the model's results, we have employed SHAP for the survival prognosis.

4.1 Data pre-processing

Before utilizing the dataset for model development, pre-processing steps were undertaken to ensure data compatibility and optimize model performance. For CT images, a series of pre-processing steps were applied. First, to facilitate feature extraction, the images were converted from the native DICOM format to RGB (Red-Green-Blue) representation. This conversion allowed the incorporation of color information, enabling the model to capture subtle patterns in the images effectively. Subsequently, to normalize the pixel values and bring them within a standardized range, a normalization [27] process was performed. This step ensures that the model is not overly influenced by variations in pixel intensity across different images, promoting consistent and reliable predictions. Finally, the processed CT images were transformed into arrays, facilitating their integration into the model.

Regarding the clinical data, appropriate pre-processing steps were undertaken to optimize its usability in the model. Categorical variables in the clinical data, such as gender or tumor grade, were subjected to label encoding [28] to convert them into numerical representations.

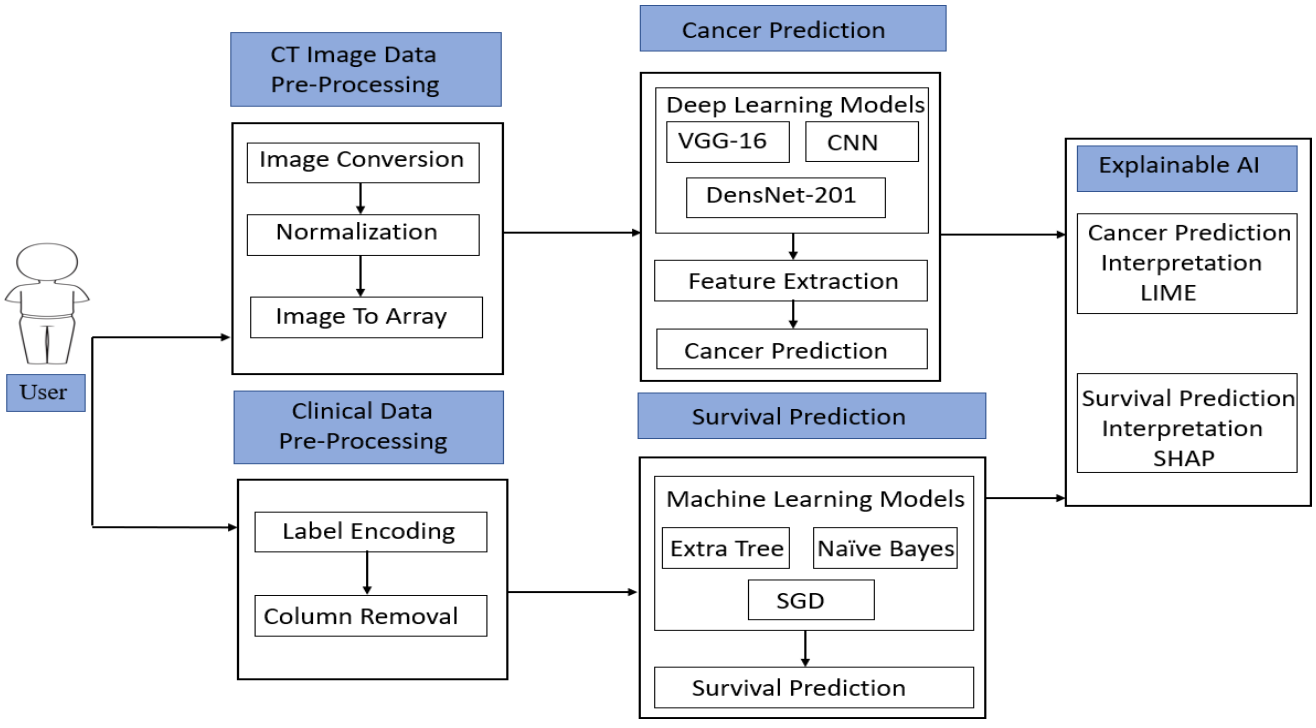


Figure 1: System architecture

This process enabled the model to work seamlessly with categorical data during the training and prediction phases. Additionally, to enhance model efficiency and eliminate irrelevant information, unwanted columns containing redundant or non-contributing features were removed from the clinical dataset. By streamlining the clinical data, the model can focus on relevant clinical factors that significantly influence the prognosis and improve the model's interpretability.

The pre-processing steps undertaken for both CT images and clinical data were essential to prepare the dataset for the subsequent model development. By ensuring data compatibility, standardization, and relevance, the interpretable analytical model can effectively leverage the combined information from CT images and clinical features to facilitate accurate pancreatic cancer detection and survival rate prediction, contributing to improved patient care and clinical decision-making.

4.2 Deep learning models and architecture

In this study, we have implemented a comprehensive framework for pancreatic cancer detection and survival prediction using state-of-the-art machine learning and deep learning models. For cancer prediction, we employed Deep learning models such as CNN, VGG-16 and DenseNet-201, which are well-suited for cancer prediction tasks involving medical imaging data like CT scans. Pancreatic cancer diagnosis relies heavily on accurately detecting subtle abnormalities in CT images. CNNs are adept at automatically learning complex features from images, making them effective in capturing intricate patterns indicative of cancerous regions in pancreatic CT scans.

A standard CNN represents a fundamental deep learning architecture, known for its simplicity and effectiveness in various image classification tasks. It serves as a baseline to compare the interpretability and performance of more complex models. VGG-16 and DenseNet-201 are deep architectures with many layers, allowing them to extract and combine rich hierarchical features, which are beneficial for the challenging task of pancreatic cancer detection. VGG-16 architecture is deeper and more sophisticated than a basic CNN. VGG-16 is a well-established model in the field of image recognition, known for its good performance and moderate complexity. DenseNet-201 is an advanced and highly deep architecture that includes dense connections between layers, enhancing feature propagation and their reuse.

By employing these three architectures, we can assess the effectiveness and consistency of interpretability methods across models of varying complexity. This enables us to understand whether these interpretability methods can provide reliable explanations for both simple and complex models, which is critical for their application in medical AI.

4.2.1 CNN model

In this study, we implemented Convolutional Neural Network model for pancreatic cancer detection using CT images as input and achieving binary yes/no predictions. The CNN architecture consists of two convolutional layers with a 3x3 filter size and 64 filters in each layer followed by a max-pooling layer with a pooling size of 2x2 to downsample the feature maps. To prevent overfitting, we applied dropout regularization with a dropout rate of 0.25 after each convolutional layer. The CNN architecture is shown in Figure 2 [21]. The learning rate was set to 0.001, and we used the Adam optimizer [29] to efficiently update the model's parameters during training.

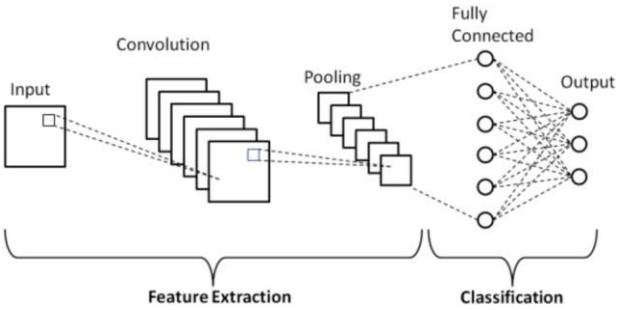


Figure 2: CNN architecture

Due to the limited size of the dataset, we decided to conduct a smaller number of epochs for training the CNN models. Specifically, we set the number of epochs to 20. The rationale behind this decision was that with a minimal dataset, the model tends to reach a saturation point in learning after a few epochs. Continuing training beyond this point led to overfitting, where the model performs well on the training data but fails to generalize to unseen data as depicted in Figure 3.

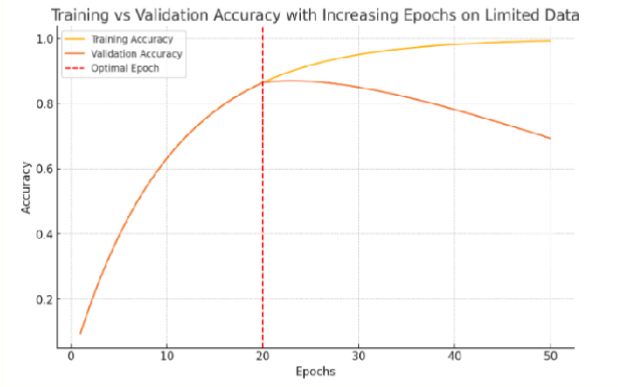


Figure 3: Epochs v/s Accuracy Graph

By fine-tuning the hyperparameters and training the CNN models with a reduced number of epochs, we aimed to strike a balance between achieving high accuracy on the training data and ensuring generalization performance on the testing data. This approach proved effective in managing the limited

dataset size and allowed us to develop reliable binary prediction models for pancreatic cancer detection using CT images as input.

4.2.2 VGG-16 model

The VGG-16 model architecture depicted in Figure 4 consists of 13 convolutional layers, each with a filter size of 3x3. The number of filters in each consecutive block varies, starting from 64 in the initial layers and progressively increasing to 128, 256, 512, and finally 512 in the deeper layers. Max-pooling is applied with a size of 2x2 to reduce spatial dimensions and extract relevant features effectively. To prevent overfitting, dropout regularization is incorporated with a rate of 0.5, randomly deactivating neurons during training to improve generalization. We used a learning rate of 0.001, a batch size of 32, and trained the model for 20 epochs. The decision to use 20 epochs was based on the limited size of the dataset and the observation that the error rate reached a stable value after these iterations. We used the Adam optimizer [29] to efficiently update the model's parameters during training.

The hyperparameters were carefully chosen to strike a balance between model performance and computational efficiency given the dataset constraints. Overall, the VGG-16 model implementation proved to be effective in generating binary yes/no predictions for pancreatic cancer detection, providing valuable insights into the potential of deep learning in early cancer diagnosis.

4.2.3 DenseNet-201

The DenseNet-201 architecture illustrated in Figure 5, is a deep convolutional neural network architecture for image recognition tasks. It employs a dense connectivity pattern where each layer receives direct input from all preceding layers, promoting feature reuse and enhancing gradient flow. The hyperparameters used for DenseNet-201 were carefully selected to achieve optimal performance. The growth rate was set to 32, ensuring the network's ability to capture relevant features effectively. We employed 6 blocks, each consisting of 48 layers, to create a deep and expressive architecture capable of capturing intricate patterns in the CT images.

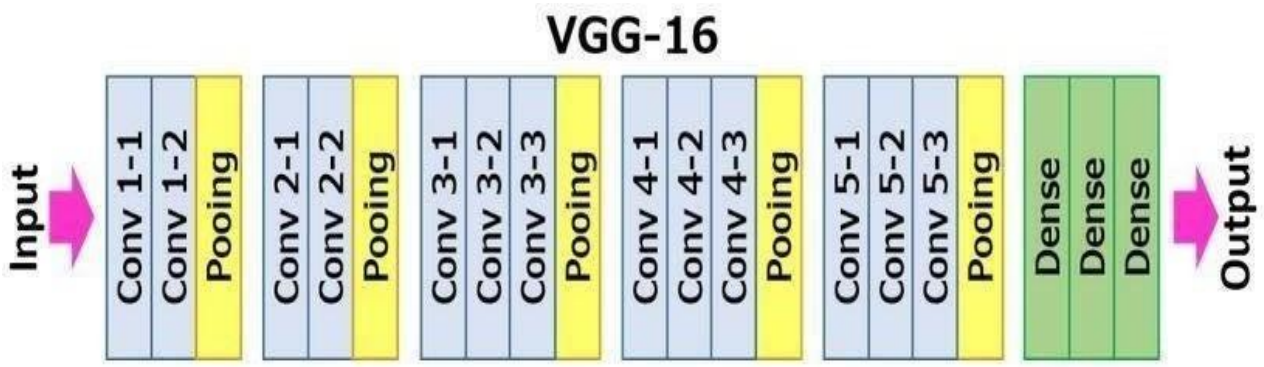


Figure 4: VGG architecture

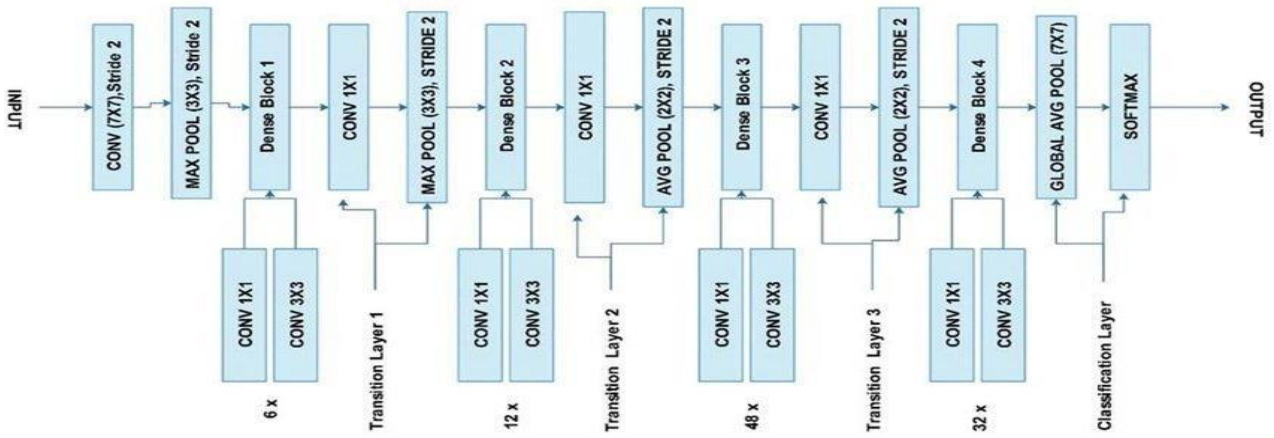


Figure 5: DenseNet-201 architecture

To prevent overfitting and improve generalization, we applied a compression factor of 0.5, which reduced the number of feature maps in transition layers. Additionally, dropout with a rate of 0.5 was introduced during training to regularize the model. We initialized the learning rate at 0.001 to facilitate a stable and effective optimization process. The choice of batch size was set to 32, balancing computational efficiency and model convergence. Considering the limited size of the dataset, we conducted training for 20 epochs, as observed that the error rate reached a near-constant level after this point. The model was trained over 20 epochs and for updating the model's parameters during the training process, Adam optimizer [29] was utilized.

In our study, hyperparameter tuning was critical for optimizing model performance. We employed grid search to systematically evaluate combinations of key hyperparameters, including learning rate, dropout rate, and batch size for all three models. Specifically, we tested learning rates of [0.0001, 0.001, 0.01], dropout rates of [0.25, 0.5], and batch sizes of [16, 32, 64]. The Adam optimizer was used across all models to ensure robust performance metrics. This approach helped us identify the optimal settings that minimized validation loss and maximized the accuracy of the models.

4.3 Machine learning models and architecture

Survival prediction in pancreatic cancer refers to the estimation of a patient's likelihood of surviving for a certain period after diagnosis, typically measured in months or years. This prediction is based on various factors, including the stage of cancer at diagnosis, patient demographics, tumor characteristics, treatment options, and other clinical data. For survival prognosis using clinical data, the choice of algorithms depends on the nature of the features and the sizeof the dataset. State-of-the-art Machine learning algorithms such as Stochastic Gradient Descent, Naïve Bayes and Extra Tree classifiers are utilized. The choice of these ML techniques is due to their diversity in model behavior.

4.3.1 Stochastic gradient descent classifier

Stochastic Gradient Descent is a widely used optimization algorithm suitable for large datasets with multiple features, as it can handle complex models effectively. This linear model is chosen for its simplicity and efficiency, particularly with large datasets. Its behavior is easy to interpret, which aligns well with the use of interpretability methods.

The goal is to estimate the chance of survival as either high or low for patients with pancreatic cancer. The SGD Regressor is a powerful machine learning algorithm that optimizes the model parameters iteratively to minimize the loss function, making it well-suited for regression tasks. To configure the SGD Regressor, we tuned several hyper-parameters to achieve the best performance. This included setting the learning rate, which controls the step size for parameter updates during training. A learning rate of 0.01 was selected to strike a balance between rapid convergence and avoiding overshooting the optimal solution. Additionally, we introduced regularization to prevent overfitting and enhance model generalization. A regularization strength of 0.1 was chosen to control the amount of regularization applied to the model. The number of iterations was set to 100 to allow the model to update its parameters gradually and converge to an optimal solution. Early stopping criteria were introduced to prevent overfitting and reduce training time. We set the early stopping threshold to 10 iterations without improvement in the loss function.

To control the type of regularization used, we applied theL2 penalty, which adds a regularization term to the loss function proportional to the square of the model weights. This helped prevent model complexity and improved generalization.

4.3.2 Naïve bayes classifier

The Naive Bayes classifier is a probabilistic algorithm that operates on the assumption of independence among the features. This means that it calculates the likelihood of each feature given the class label (chance of survival) and then combines these probabilities to determine the overall probability of survival. It can handle feature independence assumptions and is computationally efficient, making it suitable for pancreatic cancer survival prediction tasks.

Unlike other models that may involve adjusting learning rates or tuning the number of layers, the Naive Bayes classifier is inherently straightforward and computationally efficient. It is particularly useful when dealing with high-dimensional data, which can be the case in clinical datasets. As a probabilistic model, Naïve Bayes offers a different perspective compared to linear models. It assumes feature independence, making it useful for exploring how individual features contribute to predictions.

4.3.3 Extra tree classifier

Extra Tree, a type of decision tree, is advantageous when dealing with small to medium-sized datasets and categorical features. It is well-suited for pancreatic cancer survival prediction, as clinical data often includes categorical information, such as tumor grade and disease stage. This ensemble method provides a non-linear model that can capture complex interactions between features. It's particularly useful in scenarios where the relationships between features are not straightforward. Applying interpretability methods to this model allows for the exploration of how feature interactions influence predictions. The model utilized 100 estimators for ensemble learning, employed "auto" as the maximum number of features considered per split, used "Gini" as the criterion for splitting internal nodes, and allowed unlimited tree depth. It also required a minimum of 2 samples for node splitting and 1 sample for leaf nodes. With these well-optimized configurations, the Extra Tree classifier effectively predicted survival chances using

clinical data, aiding personalized treatment planning and providing reliable prognostic insights for pancreatic cancer patients.

4.4 Explainable artificial intelligence

Explainable Artificial Intelligence (XAI) [4] refers to a set of techniques and methods that aim to make complex AI models more transparent and interpretable to humans. The need for XAI arises because many advanced AI models, such as deep learning neural networks, are often considered "black boxes," [30] meaning their decision-making process is difficult for humans to understand. XAI techniques provide insights into how the model arrives at its predictions, enabling users, such as medical professionals, to trust and interpret the model's results.

For Pancreatic Cancer prediction and survival prognosis, LIME is specifically chosen for region mapping because it can help highlight the important image regions influencing the CNN model's cancer prediction. In the context of CT images for pancreatic cancer, LIME can identify the specific regions in the scan that are crucial in the CNN's decision-making process, allowing radiologists to verify the model's focus on potential cancerous areas. This transparency aids in understanding the model's strengths and limitations, contributing to improved trust and confidence in the AI-driven diagnostic tool.

On the other hand, SHAP is employed to determine the most affected features in the machine learning models used for survival prediction. SHAP provides a global explanation of the model's feature importance, which is valuable in understanding how specific clinical factors impact the model's prognostic outcomes. For pancreatic cancer, SHAP can reveal which clinical features, such as age, tumor grade, or stage, significantly influence the model's prediction of patient survival rates. This information empowers clinicians to focus on key factors in personalized treatment planning and patient care, making SHAP an essential XAI technique for survival prognosis in pancreatic cancer management.

4.4.1 Local interpretable model-agnostic explanation

The deep learning model's output, which represents the cancer likelihood of each CT image, served as the input for the LIME algorithm. For each image in the dataset, LIME generated perturbations by introducing small changes to the input image while keeping the rest of the image unchanged. After creating these perturbations, we passed them through the deep learning model to obtain their corresponding cancer likelihood predictions. LIME then weighed these perturbations based on their proximity to the original image and the similarity of the predictions. By constructing a local linear model that approximates the behavior of the complex deep learning model in the vicinity of the input image, LIME extracted the coefficients to determine the impact of different regions of the CT image on the final cancer likelihood prediction. Through this process, LIME generated region mapping by highlighting the specific areas in the input CT image that had the most significant impact on the deep learning model's prediction. These regions of interest provided valuable insights into the image features and patterns influencing the model's decision-making process for cancer detection. The LIME-generated region mapping enhanced the interpretability of the deep learning model, allowing medical professionals to validate and gain confidence in its performance for pancreatic cancer detection. This transparent and interpretable explanation facilitated a more informed and reliable diagnostic process, ultimately contributing to improved patient care in the realm of pancreatic cancer management.

4.4.2 Shapley additive explanation

For each machine learning model prediction, we use the SHAP algorithm to explain the model's output. SHAP provides us with insights into the contribution of each clinical feature to survival prediction. By quantifying the impact of each feature, SHAP helps us identify the most influential clinical parameter affecting the patient's chance of survival. In the interpretation stage, the SHAP values for each feature are plotted or presented in descending order of importance. This visualization allows medical professionals to quickly grasp the most significant clinical factor influencing survival prediction. Furthermore, the SHAP values can be used to generate summary plots or individual feature importance profiles, enabling a comprehensive understanding of how different clinical parameters contribute to the overall survival prognosis. The integration of SHAP in our machine learning model ensures that we not only obtain accurate survival predictions but also gain valuable insights into the key clinical factors driving these predictions [31]. This interpretable approach empowers healthcare practitioners to make informed decisions and tailor treatment strategies for individual patients, ultimately enhancing the overall patient care and management of pancreatic cancer.

5 Experimental analysis and results

5.1 Experimental setup

The system is designed to provide accurate cancer prediction using deep learning models CNN, VGG-16 and DenseNet-201, as well as survival prediction using machine learning models SGD, Extra Tree, and Naïve Bayes. The input data consists of CT images for cancer prediction and clinical data for survival prediction. To ensure model interpretability, the system incorporates LIME and SHAP algorithms. The front-end interface is built using HTML and CSS, allowing users to interact with the system seamlessly. The integration of the models is achieved through a Flask application, enabling easy deployment and accessibility. The coding and development of the models are carried out using Jupyter Notebook for efficient prototyping and experimentation. All the experimental cases are developed in Python in a congested environment using Anaconda tools. The

competing classification approach and various feature extraction techniques are also used, and the system is configured with an Intel Core i5-6200U processor runningat 2.30 GHz and 8GB of RAM.

5.2 Interpretation of CNN model

The CNN model effectively identified the CT image as cancer-positive with an accuracy of 0.92, indicating a potential malignancy. To interpret and validate the CNN model's prediction, LIME generated region markings on the CT image, highlighting the specific areas that influenced the model's decision, thereby offering transparency and interpretability as shown in Figure 6. The marked regions indicated the regions of interest that the model relied upon to classify the image as cancer positive. The region mapping may indicate the presence of abnormal or suspicious structures within the pancreas or surrounding tissues, such as tumors, lesions, or other cancerous features. These marked regions could correspond to distinct patterns, shapes, or textures that the CNN model learned to associate with cancerous areas in the CT images. Similarly, the CNN model effectively identified the CT image as cancer-negative, indicating a potential malignancy. To interpret and validate the CNN model's prediction, LIME generated region markings on the CT image, highlighting the specific areas that influenced the model's decision, thereby offering transparency and interpretability.

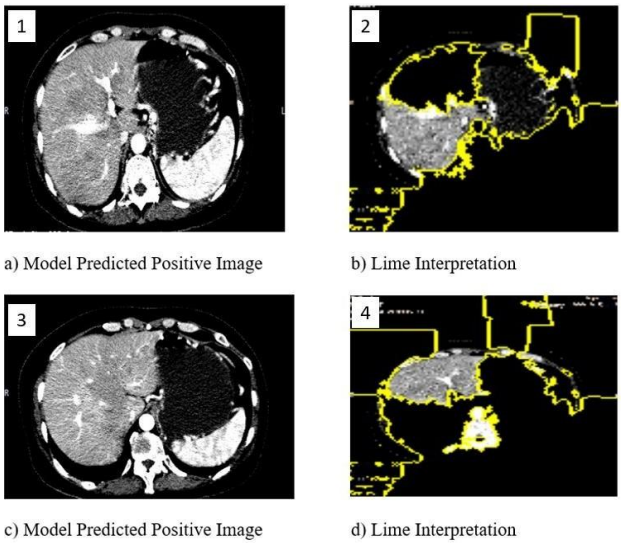


Figure 6: CNN interpretation for positive image

5.3 Interpretation of DenseNet-201 model

The interpretation provided by the DenseNet-201 model in region mapping using the LIME algorithm is potentially superior to that of a traditional CNN model. This is due to DenseNet's unique architecture and skip connections which enable more efficient feature propagation and deeper layer utilization. DenseNet-201 is a densely connected neural network that incorporates skip connections, allowing direct connections between all layers within the network. This dense connectivity facilitates the propagation of gradients and information throughout the network, leading to better feature reuse and representation. In the context of region mapping, this means that the DenseNet-201 model can capture a more comprehensive and intricate understanding of the CT images' features, including subtle patterns and contextually relevant information. As a result, when LIME performs region mapping using the DenseNet-201 model, it can identify and highlight even more specific and relevant regions within the CT images thatinfluenced the model's prediction. The dense connectivity and feature reuse in DenseNet-201 allow for more precise localization of the regions of interest, potentially offering a more detailed and accurate interpretation of the areas that are indicative of cancerous regions. Similarly, the DenseNet-201 model effectively identified the CT image as cancer-negative, indicating a potential malignancy. To interpret and validate the DenseNet-201 model's prediction, LIME generated region markings on the CT image, highlighting the specific areas that influenced the model's decision, thereby offering transparency and interpretability.

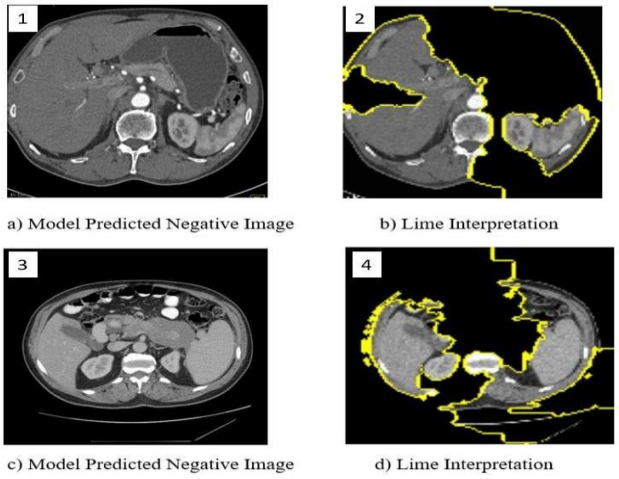


Figure 7: CNN interpretation for negative image

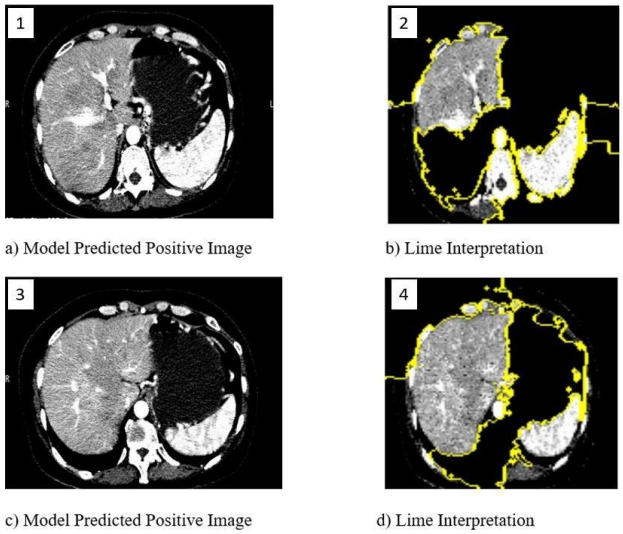


Figure 8: DenseNet-201 interpretation for positive image

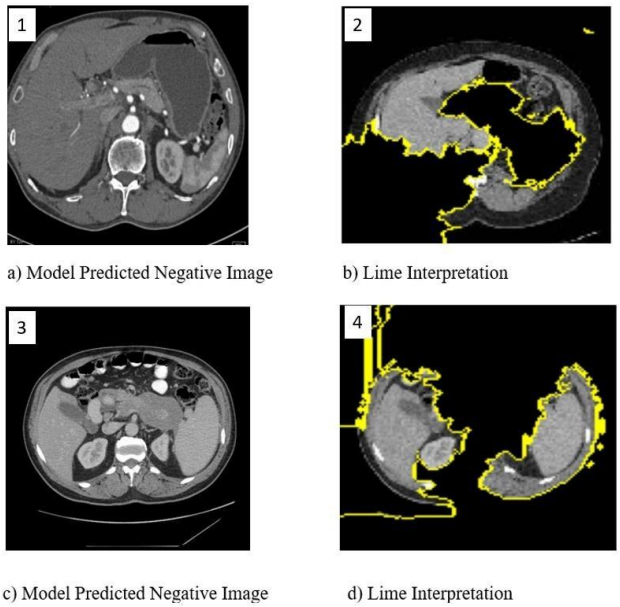


Figure 9: DenseNet-201 interpretation for negative image

5.4 Interpretation of VGG-16 model

The VGG-16 model demonstrated commendable predictive capabilities, accurately identifying the CT image as cancer-positive, which highlights its proficiency in distinguishing cancerous regions from non-cancerous ones. However, when comparing VGG-16's interpretability in region mapping through the LIME algorithm with the results obtained from DenseNet-201, we observed that VGG-16 exhibited some limitations. While VGG-16 effectively provided relevant region markings, it showed relatively poorer results compared to DenseNet-201 in precisely localizing intricate patterns and subtle features associated with pancreatic cancer.

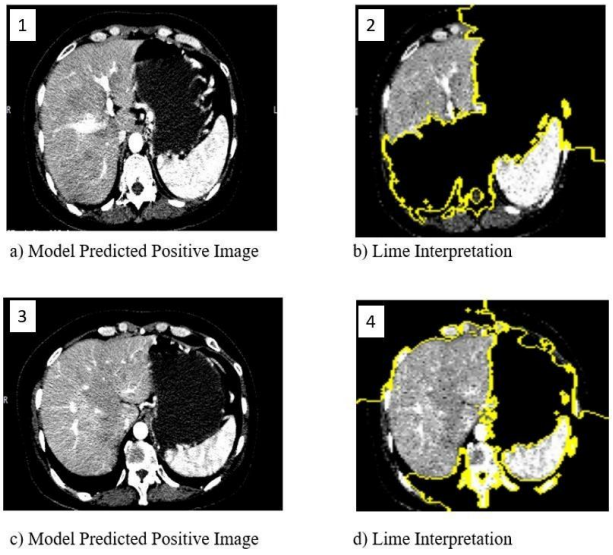


Figure 10: VGG-16 interpretation for positive image

The difference in performance can be attributed to the architectural dissimilarity between VGG-16 and DenseNet-201. DenseNet-201's densely connected structure allows for better feature reuse and a more holistic understanding of the CT images, enabling a more detailed and accurate region mapping through LIME. In contrast, VGG-16's deeper architecture with a higher number of parameters may have led to limited feature reuse and potentially diminished sensitivity to specific cancerous regions. Similarly, the VGG-16 model effectively identified the CT image as cancer-negative, indicating a potential malignancy. To interpret and validate the VGG-16 model's prediction, LIME generated region markings on the CT image, highlighting the specific areas that influenced the model's decision, thereby offering transparency and interpretability.

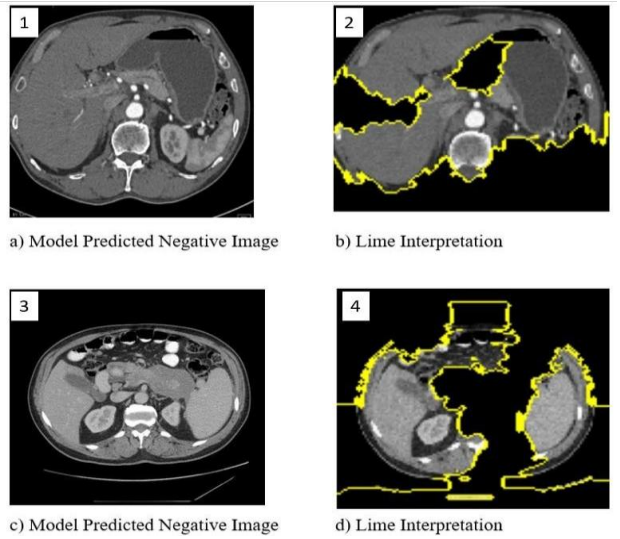


Figure 11: VGG-16 interpretation for negative image

Overall, in the context of pancreatic cancer detection using CT images, the DenseNet-201 model outperformed both the CNN and the VGG-16 models. DenseNet-201 demonstrated superior predictive capabilities, providing accurate and reliable cancer-positive predictions. Additionally, DenseNet-201 showed better interpretability through the LIME algorithm, delivering more precise and detailed region mappings compared to VGG-16 and CNN. The dense connectivity and feature reuse of DenseNet-201 enabled it to capture intricate patterns and contextually relevant information in CT images, leading to a more comprehensive and accurate localization of cancerous regions.

LIME provided the best region mapping results when combined with the DenseNet-201 model. The superior region mapping with LIME and DenseNet-201 can be attributed to the unique architecture of DenseNet-201, which facilitates better feature reuse and representation. This, in turn, allowed LIME to identify the most relevant and influential regions in the CT images, providing a comprehensive view of the features contributing to cancer prediction. The precise region mapping obtained from LIME with DenseNet-201 enhanced interpretability, instilling greater trust in the model's predictions, and enabling medical professionals

to validate and comprehend the decision-making process with confidence.

The model's significance lies in its potential to revolutionize pancreatic cancer management in society. By combining the accurate and interpretable predictions from DenseNet-201 and LIME, our model provides medical practitioners with a reliable AI-driven diagnostic tool. The model's transparency and interpretability through LIME facilitate better understanding and validation of predictions, fostering more informed clinical decision-making. Additionally, the efficient integration of deep learning and XAI techniques optimizes diagnostic workflows, reducing medical costs, and improving access to timely and accurate cancer diagnosis for a broader population. Ultimately, this will improve patient outcomes, supporting medical professionals in their mission to combat pancreatic cancer, and making a positive impact on society's healthcare landscape.

Evaluation methods: In the performance assessment of our experiment, we use four crucial evaluation metrics: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). True Positive represents the number of cases correctly predicted as positive or cancerous instances. False Positive indicates the number of cases wrongly predicted as positive when they are actually negative or non-cancerous. True Negative reflects the number of cases correctly predicted as negative or non-cancerous instances. False Negative, on the other hand, represents the number of cases incorrectly predicted as negative when they are actually positive or cancerous. These four measurements are fundamental in understanding the model's predictive accuracy and its capability to correctly classify cancer-positive and cancer-negative instances in pancreatic cancer detection

Accuracy = (TP+TN) / (TP+TN+FP+FN)
Precision = TP / (TP+FP)
Recall = TP / (TP+FN)
F1 Score = 2× (Precision *Recall) / (Precision +Recall)

Table 2: Performance analysis

DL Model	Accuracy	Precision	Recall	F1-Score
CNN	0.92	0.92	0.91	0.93
VGG-16	0.93	0.92	0.95	0.93
DenseNet-201	0.95	0.93	0.97	0.96

The comparison of accuracy among the three algorithms is provided in Table 2. DenseNet-201 emerges as the most accurate model for pancreatic cancer detection using CT images, achieving an impressive accuracy of 0.95. DenseNet-201's dense connectivity and skip connections allow it to effectively capture and reuse features, leading to a more comprehensive understanding of the intricate patterns indicative of pancreatic cancer. In close competition, VGG-16 also demonstrates strong performance with an accuracy of 0.93, showcasing its capability to distinguish cancerous and non-cancerous regions in CT images. While the CNN model performs well with an accuracy of 0.92, it falls slightly behind DenseNet-201 and VGG-16 in predictive power. These results highlight the superiority of deep learning architectures, particularly DenseNet-201, in accurately identifying pancreatic cancer using CT images, reinforcing their potential as valuable tools in the early diagnosis and management of this challenging disease.

Further, to determine the statistical significance of the differences in the predictive accuracies of the three models, one-way ANOVA statistical significance test is conducted. The ANOVA (Analysis of Variance) test checks whether there are statistically significant differences between the means of the different groups (in this case, CNN, DenseNet-201, and VGG-16). ANOVA returns a F-statistic of 6.2419 and p-value of 0.0199. F-statistic of 6.2419 indicates a notable variance between the models' accuracy (CNN, VGG-16, DenseNet-201) compared to the variance within each group. The p-value of 0.0199, being less than 0.05, confirms that the differences in accuracy between the models are statistically significant overall. Further analysis through post-hoc tests, such as Tukey HSD is performed, to pinpoint which specific pairs of models differ. Tukey HSD results are shown in Table 3.

According to Tukey's HSD test, DenseNet-201 has a significantly higher accuracy compared to CNN ($p = 0.0166$), while there is no significant difference between CNN and VGG-16 ($p = 0.4065$) or between DenseNet-201 and VGG-16 ($p = 0.1333$). This suggests that DenseNet-201 is the best-performing model among the three.

The graph in Figure 12 illustrates the Receiver Operating Characteristic (ROC) curves for three different models: CNN, VGG-16, and DenseNet-201. Each curve effectively plots the true positive rate against the false positive rate, showcasing the models' performance across various classification thresholds. The area under the curve (AUC) is indicated for each model, with CNN achieving an AUC of 0.52, VGG-16 at 0.51, and DenseNet-201 at 0.57.

These values demonstrate that all three models exhibit competitive discriminatory power, with DenseNet-201 showing a slight edge over the others. The results indicate that the models can provide valuable insights into the classification task at hand. The ROC curves serve as an effective visual tool for assessing model performance, highlighting the potential for further refinement and optimization of these models to enhance their predictive capabilities in future work.

Table 3: Results of Tukey HSD test

Group1	Group2	Mean Difference	P-adj	Lower	Upper	Reject
CNN	DenseNet-201	0.0325	0.0166	0.0066	0.0584	True
CNN	VGG-16	0.0125	0.4065	-0.0134	0.0384	False
DenseNet-201	VGG-16	-0.02	0.1333	-0.0459	0.0059	False

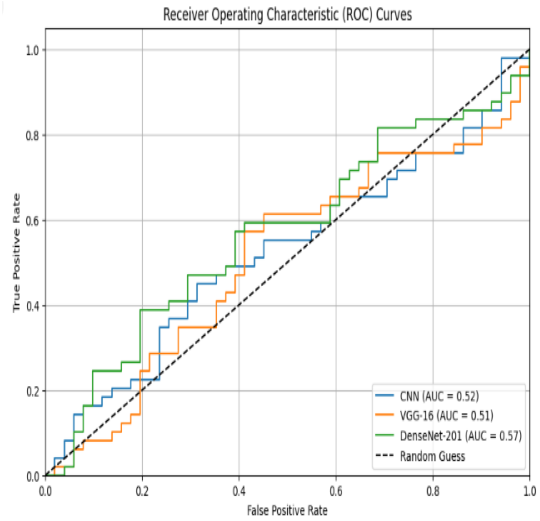


Figure 12: Receiver operating characteristic curve

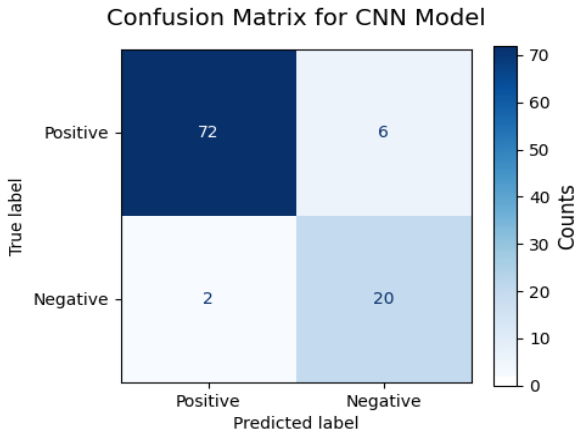


Figure 13: Confusion matrix for CNN.

The confusion matrices for the CNN, VGG-16 and DenseNet-201 are shown in Figures 13, 14 and 15 respectively providing insights into their classification performance. Each matrix shows the counts of true positives, true negatives, false positives, and false negatives for the respective models. These values help assess the accuracy and errors made during prediction, with true positives and negatives indicating correct classifications, while false positives and negatives highlight misclassifications.

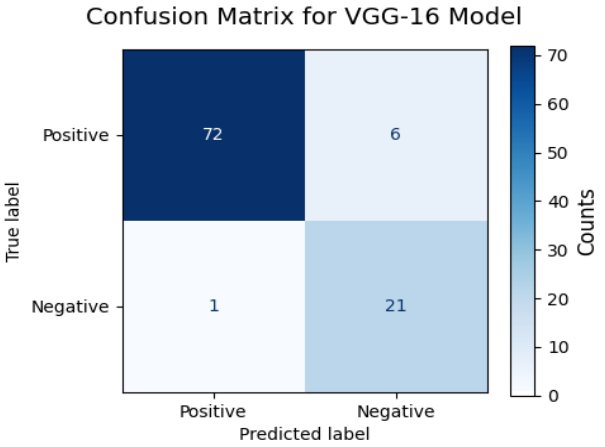


Figure 14: Confusion matrix for VGG-16.

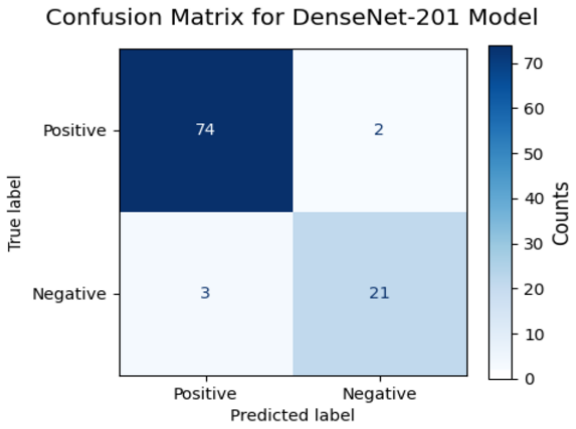


Figure 15: Confusion matrix for DenseNet-201

5.5 Analysis of survival prognosis

The selection of features for survival prediction in pancreatic cancer is a crucial step in building an accurate and meaningful model. The chosen features KI-67 Index, PFS Months, Creatine, Age, Tumor Grade, CA 19/9 U/ml, and Gender, have been carefully selected based on their clinical relevance and potential impact on patient survival outcomes [32].

KI-67 Index: KI-67 is a protein that is closely associated with cell proliferation and tumor growth. Its measurement provides valuable information about the rate of tumor cell division, which is a critical factor in predicting tumor aggressiveness and patient survival.

Table 4: Selected features

KI-67 Index
PFS Months
Creatine
Age
Tumor Grade
CA 19/9 U/ml
Gender

PFS months (Progression-Free Survival Months): PFS is a critical clinical endpoint in cancer prognosis, representing the time from the start of treatment to disease progression or relapse. It is a strong indicator of treatment efficacy and overall survival.

Creatine: Creatine levels in the blood can be indicative of kidney function, which is essential for monitoring potential complications and overall health status during cancer treatment.

Age: Age is a significant prognostic factor in cancer outcome. Younger patients may have a better overall health status and tolerate treatments more effectively, while older patients may have additional comorbidities that influence their survival.

Tumor grade: Tumor grade is a measure of tumor cell differentiation and aggressiveness. Higher tumor grades typically indicate more aggressive tumors that may have a poorer prognosis.

CA 19/9 U/ml: CA 19-9 is a tumor marker associated with pancreatic cancer. Elevated levels may indicate advanceddisease and a higher risk of poor outcomes.

Gender: Gender can also play a role in cancer prognosis, as some types of cancers may behave differently in males and females.

The selected features encompass a wide range of clinical and biological factors that are known to impact pancreatic cancer survival outcomes. By incorporating these diverse and relevant features, the survival prediction model can capture the complex interactions between clinical characteristics and tumor biology, leading to more accurate and personalized prognostic assessments. This feature selection process ensures that the developed model is both clinically meaningful and robust, enhancing its usefulness in guiding treatment decisions and improving patient outcomes in pancreatic cancer management.

While the other two algorithms, namely SGD and Naive Bayes, may also provide insights into the most affected features for survival prediction in pancreatic cancer, their accuracy in determining these features is not as precise as that of the Extra Tree classifier. The less accurate identification of the most influential clinical factors in the survival prognosis by SGD and Naive Bayes might be attributed to their respective algorithmic limitations. SGD is an optimization-based method that may not fully capture intricate feature interactions, while Naive Bayes assumes independence between features, potentially overlooking complex relationships. As a result, their feature importance rankings might lack the precision exhibited by the Extra Tree model. Due to this disparity in performance, we have chosen to focus on the Extra Tree classifier as it provides a more reliable and accurate understanding of the critical clinical factors influencing patient survival in pancreatic cancer.

The three models (SGD, Naive Bayes, and Extra Tree) may not necessarily give the same "most affected feature" in survival prediction using the SHAP algorithm. The reason lies in the inherent differences in the algorithms' working principles and the way they determine feature importance.

SGD is an optimization algorithm that iteratively updates the model's parameters to minimize the prediction error. It may assign different weights to different features during this process, leading to variations in feature importance.

Naive Bayes is a probabilistic algorithm based on the Bayes theorem. It assumes independence between features given the class label. Due to this assumption, Naive Bayes may not capture complex interactions between features, and its feature importance ranking might differ from other models.

Extra Tree is an ensemble learning method that builds multiple decision trees and combines their predictions. It selects random subsets of features and nodes during tree construction, introducing randomness that can lead to different feature importance rankings.

As a result, the three models may prioritize features differently in terms of their impact on survival predictions. The differences could be more pronounced when the dataset is not large enough to provide a comprehensive view of feature interactions. Comparing the models' performance metrics, such as feature importance scores derived from the SHAP algorithm, on anindependent test dataset would help identify the model that best captures the crucial clinical factors affecting patient survival in the context of survival prognosis. Additionally, conducting feature importance analysis and clinical validation with domain experts could shed more light on the models' interpretability and reliability in survival prognosis.

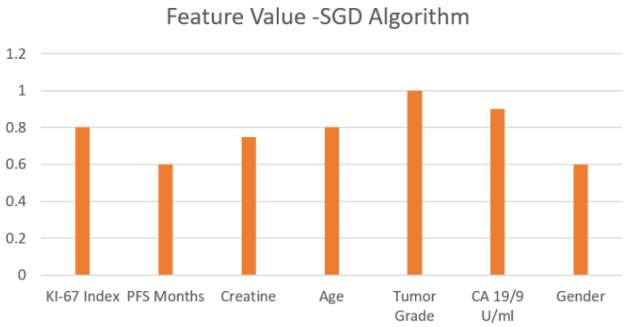


Figure 16: Feature value graph for SGD algorithm

Figure 16 displays feature importance values for the SGD algorithm in predicting survival outcomes for pancreatic cancer patients. Tumor grade holds the highest importance with a value of 1.0, followed by CA 19/9 U/ml (0.9) and KI-67 Index (0.8). Age, Creatine, and

PFS Months show importance values of 0.8, 0.7, and 0.6, respectively.

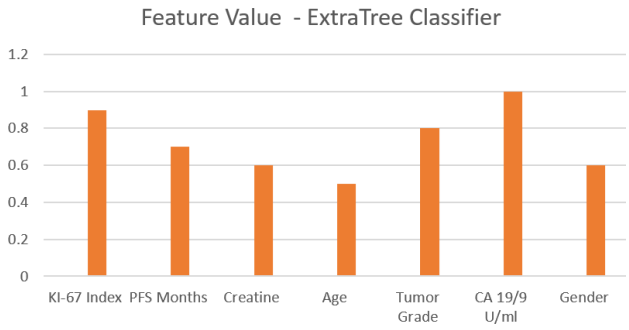


Figure 17: Feature value graph for extratree algorithm

The ExtraTree algorithm's feature importance graph as shown in Figure 17 illustrates the relative significance of each feature in predicting survival outcomes for pancreatic cancer patients. CA 19/9 U/ml emerges as the most important feature with a value of 1.0, signifying its critical role in determining patient survival. KI-67 Index closely follows with an importance value of 0.85, underscoring its considerable impact on survival predictions. Tumor Grade ranks third with a value of 0.8, indicating its significant contribution to survival prognosis. Additionally, PFS Months and Creatine, demonstrate importance values of 0.7 and 0.6, respectively, highlighting their relevance in predicting patient outcomes. Age exhibits the lowest importance with a value of 0.45, suggesting its relatively minor influence on survival predictions.

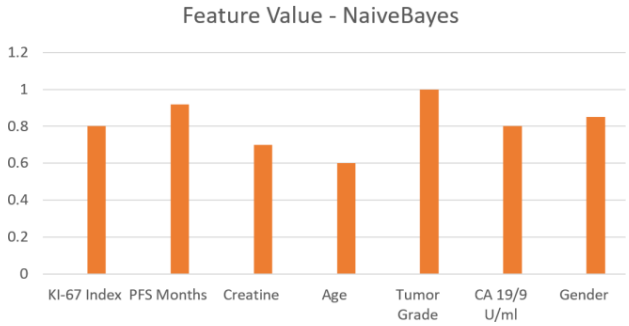


Figure 18: Feature value graph for naïve bayes algorithm

The feature importance graph as shown in Figure 18 for the Naive Bayes algorithm reveals the relative significance of each feature in predicting survival outcomes for pancreatic cancer patients. Tumor grade holds the highest importance with a value of 1.0, followed by PFS Months (0.85), CA 19/9 U/ml (0.8) and KI-67 Index (0.8). Age, Creatine, and PFS Months show lower importance scores.

In conclusion, based on the comparison of feature importance and interpretability using the SHAP algorithm, the Extra Tree classifier stands out as the top-performing model for survival prediction in pancreatic cancer. Its ability to accurately identify the most affected features highlights its potential as a valuable tool for medical professionals in personalized treatment planning and decision-making for patients with pancreatic cancer.

6 Conclusion and future work

In this comprehensive study, we investigated a diverse range of algorithms for pancreatic cancer prediction and survival prognosis. Our findings demonstrate that the VGG-16 and DenseNet-201 outperformed CNN algorithms in cancer prediction, achieving high accuracy in distinguishing cancer-positive and cancer-negative CT images. The adoption of LIME for model validation provided interpretable insights by marking the regions influencing the models' predictions, fostering trust and facilitating the integration of AI-driven diagnostics into clinical workflows.

For survival prediction, the machine learning models SGD, Extra Tree, and Naïve Bayes exhibited promising performance, with the SHAP algorithm delivering crucial insights into the most impactful clinical factors affecting patient outcomes. SHAP's ability to explain model predictions enabled medical practitioners to identify key prognostic indicators, empowering them to make informed decisions and improve patient care.

The utilization of Explainable Artificial Intelligence techniques, such as LIME and SHAP, ensured that both cancer prediction and survival prognosis were transparent and interpretable. This interpretability not only boosts the confidence of medical professionals in the model's predictions but also provides a deeper understanding of the intricate decision-making processes underlying each model. The study has significant implications for society by combining powerful AI-driven algorithms with interpretable XAI techniques, medical practitioners can confidently make data-driven decisions, leading to improved patient outcomes and survival rates.

Future work should focus on expanding the dataset to include more diverse cases and increasing the sample size to improve model generalization. Additionally, integrating other relevant data sources, such as genomic data or histopathological features, could lead to a more comprehensive and accurate cancer prediction system. Further research could explore the combination of multiple imaging modalities and clinical data to gain a deeper understanding of the disease. Moreover, incorporating other XAI techniques or developing hybrid interpretability methods might provide even greater insights into model predictions, boosting confidence and facilitating widespread clinical adoption. Finally, conducting rigorous validation studies on real-world patient data and collaborating with medical practitioners for clinical validation will be crucial for the successful translation of this framework into clinical practice.

Disclosure Statement

Statements of ethical approval

Not applicable

Funding
This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profitsectors.

Competing Interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials / Code:
Data and code will be made available on request.

References

[1] K. Honda, Y. Hayashida, T. Umaki, T. Okusaka, T. Kosuge, S. Kikuchi, and F. Moriyasu. Possible detection of pancreatic cancer by plasma protein profiling. Cancer Res., vol. 65, no. 22, pp. 10613-10622. 2005.

https://doi.org/10.1158/0008-5472.can-05-1851

[2] P. Maisonneuve and A. Lowenfels. Epidemiology of Pancreatic Cancer: An Update. Digestive Diseases (Basel, Switzerland), vol. 28, pp. 645-656. 2010.

https://doi.org/10.1159/000320068

[3] Si, Ke, Ying Xue, Xiazhen Yu, Xinpei Zhu, Qinghai Li, Wei Gong, Tingbo Liang, and Shumin Duan. "Fully end-to-end deep-learning-based diagnosis of pancreatic tumors." Theranostics 11, no. 4: 1982, 2021.

https://doi.org/10.7150/thno.52508

[4] F. K. Došilović, M. Brčić and N. Hlupić. Explainable artificial intelligence: A survey. 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia. pp. 0210-0215, 2018.

http://dx.doi.org/10.23919/MIPRO.2018.8400040

[5] Wei Xuan, Guang qiang You. Detection and diagnosis of pancreatic tumor using deep learning-based hierarchical convolutional neural network on the internet of medical things platform. Future Generation Computer Systems, Volume 111, pp. 132-142, 2020. ISSN 0167-739X. https://doi.org/10.1016/j.future.2020.04.037.

[6] Z. Zhang, S. Li, Z. Wang, and Y. Lu. A Novel and Efficient Tumor Detection Framework for Pancreatic Cancer via CT Images. 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, pp. 1160-1164, 2020.

https://doi.org/10.1109/EMBC44109.2020.9176172

[7] Wei Zhao, Liyue Shen, Bin Han, Yong Yang, Kai Cheng, Diego A.S. Toesca, Albert C. Koong, Daniel T. Chang, Lei Xing. Marker less Pancreatic Tumor Target Localization Enabled by Deep Learning. International Journal of Radiation Oncology*Biology*Physics, Volume 105, Issue 2, pp. 432-439, 2019, ISSN 0360-3016.

https://doi.org/10.1016/j.ijrobp.2019.05.071.

[8] B. S. Hameed and U. M. Krishnan. Artificial Intelligence- Driven Diagnosis of Pancreatic Cancer. Cancers, vol. 14, no. 21, pp. 5382, 2022.

https://doi.org/10.3390/cancers14215382

[9] Li, M., Nie, X, Reheman, Y., Huang, P., Zhang, S., Yuan, Y., & Han, W. Computer-aided diagnosis and staging of pancreatic cancer based on CT images. IEEE Access, 8, pp. 141705-141718, 2020. https://doi.org/10.1109/access.2020.3012967

[10] D. Agarwal, O. Covarrubias-Zambrano, S. H. Bossmann and B. Natarajan. Early Detection of Pancreatic Cancers Using Liquid Biopsies and Hierarchical Decision Structure. IEEE Journal of Translational Engineering in Health and Medicine, vol. 10, pp. 1-8, 2022.

https://doi.org/10.1109/jtehm.2022.3186836

[11] Sekaran, K., Chandana, P., Krishna, N. M., & Kadry, S.. Deep learning convolutional neural network (CNN) With Gaussian mixture model for predicting pancreatic cancer. Multimedia Tools and Applications, 79 (15), pp. 10233-10247, 2020.

https://doi.org/10.1007/s11042-019-7419-5

[12] Srinidhi, B., and M. S. Bhargavi. An XAI Approach to Predictive Analytics of Pancreatic Cancer. International Conference on Information Technology (ICIT), pp. 343-348, 2023.

https://doi.org/10.1109/icit58056.2023.10225991

[13] Bobes-Bascarán, José, Eduardo Mosqueira-Rey, Ángel Fernández-Leal, Elena Hernández-Pereira, David Alonso-Ríos, Vicente Moret-Bonillo, Israel Figueirido-Arnoso, and Yolan da Vidal-Ínsua. Evaluating Explanatory Capabilities of Machine Learning Models in Medical Diagnostics: A Human-in-the-Loop Approach. arXiv preprint arXiv:2403.19820, 2024

[14] Bobes-Bascarán, José, Ángel Fernández-Leal, Eduardo Mosqueira-Rey, David Alonso Ríos, Elena Hernández-Pereira, and Vicente Moret-Bonillo. Understanding Machine Learning Explainability Models in the context of Pancreatic Cancer Treatment. VI Congreso Xove TIC: impulsando el talento científico. A Coruña, pp. 175-182. Universidade da Coruña, Servizo de Publicacións, 2023.

https://doi.org/10.17979/spudc.000024.28

[15] Goel, Siya. Using Feature Selection and Ensemble Algorithms for Sophisticated Pancreatic Cancer Diagnosis. HAL openscience, hal-03119399f, 2021.

[16] Bakasa, Wilson, and Serestina Viriri. Pancreatic Cancer Survival Prediction: A Survey of the State-of-the-Art. Computational and Mathematical

Methods in Medicine, vol. 2021, pp. 1–17, 2021. https://doi.org/10.1155/2021/1188414

[17] Baek, Bin, and Hyunju Lee. Prediction of survival and recurrence in patients with pancreatic cancer by integrating multi-omics data. Scientific reports, vol. 10, no. 1, 2020. https://doi.org/10.1038/s41598-020-76025-1

[18] Keyl, Julius, Stefan Kasper, Marcel Wiesweg, Julian Götze, Martin Schönrock, Marianne Sinn, Aron Berger et al.. Multimodal survival prediction in advanced pancreatic cancer using machine learning. ESMO open 7, no. 5: 100555, 2022. https://doi.org/10.1016/j.esmoop.2022.100555

[19] Bertsimas, Dimitris, Georgios Antonios Margonis, Yifei Huang, Nikolaos Andreatos, Holly Wiberg, Yu Ma, Caitlin Mcintyre et al. Toward an Optimized Staging System for Pancreatic Ductal Adenocarcinoma: A Clinically Interpretable, Artificial Intelligence–Based Model. JCO clinical cancer informatics 5: pp. 1220-1231, 2021. https://doi.org/10.1200/cci.21.00001

[20] The Cancer Imaging Archive https://www.cancerimagingarchive.net/

[21] Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017. https://doi.org/10.1145/3065386

[22] Tammina, S. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. International Journal of Scientific and Research Publications (IJSRP), vol. 9, no. 10, p. p9420, 2019. https://doi.org/10.29322/ijsrp.9.10.2019.p9420

[23] Y. Shi, T. Zhou, Z. You, J. Chu and T. Li. AM-DenseNet: A Novel DenseNet Framework using Attention Mechanisms for COVID-19 CT Image Classification. IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS), Chengdu, China, pp. 474-479, 2022. https://doi.org/10.1109/ccis57298.2022.10016401

[24] T. Kobayashi. SCW-SGD: Stochastically Confidence- Weighted SGD. IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, pp. 1746-1750, 2020. https://doi.org/10.1109/icip40778.2020.9190992

[25] B. Dhananjay, N. P. Venkatesh, A. Bhardwaj and J. Sivaraman. Cardiac signals classification based on Extra Trees model. 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, pp. 402-406, 2021. https://doi.org/10.1109/spin52536.2021.9565992

[26] V. Vijay and P. Verma. Variants of Naïve Bayes Algorithm for Hate Speech Detection in Text Documents. International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, pp. 18-21, 2023. https://doi.org/10.1109/aisc56616.2023.10085511

[27] M. Kolarik, R. Burget and K. Riha. Comparing Normalization Methods for Limited Batch Size Segmentation Neural Networks. 43rd International Conference on Telecommunications and Signal Processing (TSP), Milan, Italy, pp. 677-680, 2020. https://doi.org/10.1109/tsp49548.2020.9163397

[28] R. Guedrez, O. Dugeon, S. Lahoud and G. Texier. Label encoding algorithm for MPLS Segment Routing. 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 2016, pp. 113-117, 2016. https://doi.org/10.1109/nca.2016.7778603

[29] N. A. M. Ariff and A. R. Ismail. Study of Adam and Adamax Optimizers on AlexNet Architecture for Voice Biometric Authentication System. 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Republic of Korea, pp. 1-4, 2023. https://doi.org/10.1109/imcom56909.2023.10035592

[30] P. Rasouli and I. C. Yu. Explainable Debugger for Black-box Machine Learning Models. International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, pp. 1-10, 2021. https://doi.org/10.1109/ijcnn52387.2021.9533944

[31] J. Rebane, I. Samsten, P. Pantelidis and P. Papapetrou. Assessing the Clinical Validity of Attention-based and SHAP Temporal Explanations for Adverse Drug Event Predictions. IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS), Aveiro, Portugal, pp. 235-240, 2021. https://doi.org/10.1109/cbms52027.2021.00025

[32] H. Gong, P. Zuliani, Q. Wang and E. M. Clarke. Formal analysis for logical models of pancreatic cancer. 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, pp. 4855-4860, 2011. https://doi.org/10.1109/cdc.2011.6161052

https://doi.org/10.31449/inf.v48i4.5496 Informatica **48** (2024) 641–652 **641**

Wavelet Decompositions, Hierarchical Encoding and Convolutional Neural Network Integrated Lossless Audio Codec

Asish Debnath, Uttam Kr. Mondal
Dept. of Computer Science, Vidyasagar University, Midnapore, 721102, West Bengal, India
E-mail: debnathasish@gmail.com, uttam_ku_82@yahoo.co.in

Keywords: CNN, MSE, MAE, lossless audio compression, approximation coefficients, detail coefficients, hierarchical encoding

Received: November 30, 2023

In this paper, a lossless audio codec is proposed by leveraging Wavelet transformation, Hierarchical encoding with Convolutional Neural Network architecture. In the first phase, three level 1D wavelet decomposition is applied on the input audio for generating approximation and detail coefficients. In the next phase, the approximation and detail coefficients are transformed into binary streams by utilizing the proposed dynamic hierarchical encoding algorithm. In this encoding technique, coefficients are converted to binary by dynamically accumulating the binary path values. In the subsequent phase, the binary stream is transformed into image patterns and further compressed by reducing the dimensionality by the proposed convolutional neural network(CNN) model. The model's effectiveness is evaluated against current conventional lossless audio benchmarks and machine learning-based methods. Experiment results demonstrate that the method shows better performance than existing lossless audio techniques.

Povzetek: Razvit je avdio kodek, ki združuje valovne transformacije, hierarhično kodiranje in konvolucijske nevronske mreže za izboljšanje kompresije.

1 Introduction

In today's world, immense amount of audio data is being generated at every moment. Therefore, using the network bandwidth and storage space efficiently, audio data compression is one of the paramount important. The advent of deep neural network opened the possibilities of achieving excellent result along with the conventional techniques in this area. Lossless audio compression [1] is the audio compression technique utilized whenever the requirement is to preserve the quality of the original input audio and reconstructed audio signal. It also reduces the file size without losing the audio information. On the contrary, lossy audio compression losses some audio data permanently to achieve higher compression. Lossless audio data [2] compression is used where data loss is not expected at all. A graph based [3] and cluster quantization [4] based audio encoding techniques introduced recently. Deep learning [5] based approaches are applied recently in audio compression. Lossless compression is required in medical industies for compression and sending various bio-signals [6].
In this present work, we proposed an lossless audio codec (WLCLAC) by sequentially integrating three layer 1D wavelet [7] decomposition, adaptive hierarchical binary encoding, and CNN [8] compression architecture. The proposed model works in three stages. In the first step, the input audio sampled values are transformed using a three-level one-dimensional discrete wavelet decomposition approach. Using wavelet transformations, detail coefficients(cD) and approximation coefficients(cA) are generated at each level of decomposition. For the signal S, the structure of the wavelet decomposition at i level is as follows: $[cA_i, cD_i, .., cD_1]$ For i = 3, this structure comprises the terminal nodes of the following tree shown in figure 1.
In the second step, hierarchical binary encoding technique is applied on the input wavelet coefficients. Coefficients are segregated into integers first, then search in the proposed hierarchy. If the node is matched with the input digit, then binary path values are accumulated and translated as an encoded binary stream of the input digit.
In the third step, CNN encoder decoder model is used to further compress the binary stream.

Figure 2 depicts the encoding and decoding model. Utilizing the encoder, size of the input audio compressed in the encoding step. Compressed signal stored as the latent space with reduced and compressed form. Decoder section reconstructs the signal which is very much similar to the input.

The proposed technique has significant potential for real-time applications in the field of audio compression. Upon deployment, it can enhance the compression ratio and processing speed while reducing the model's complexity. This improvement would lead to reduced memory usage and lower network bandwidth requirements, making the system highly efficient for real-time audio transmission and storage in various applications, such as streaming services, communication systems, and data storage solutions.

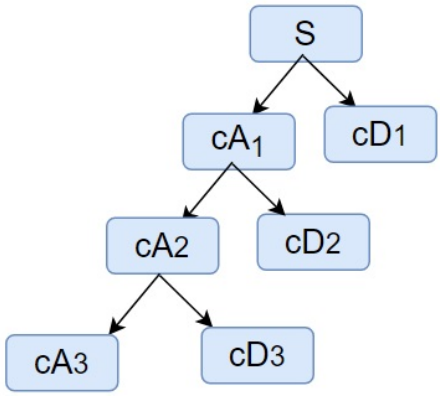


Figure 1: Three level 1D wavelet decomposition tree

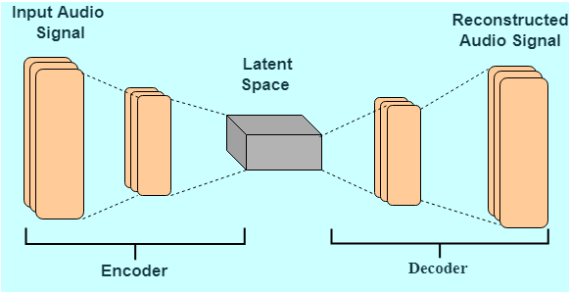


Figure 2: Encoder and decoder framework

1.1 Notations and symbols

The terminology and their complete explanations that are relevant to the abbreviations used in this work are provided in Table 1.

Table 1: Symbol and abbreviation form of some terms

Symbol / Abbreviation	Meaning / Full form
CNN	Convolutional Neural Network
DNN	Deep neural networks
PNSR	Peak signal-to-noise ratio
NCC	Normalised cross correlation
cA	Approximation coefficients
cD	Detail coefficients

2 Literature survey

In 2003, lossless audio codec standard MPEG-4(ALS) [9] was introduced. This LPC (linear prediction encoding) based technique improves the residual coding and reduce the bit rate compared to PCM like approaches. The predictor coefficient cost during decoding and demultiplexing is this codec's drawback. A new variant of this technique, MPEG-4 ALS (RLS-LMS) introduced later by combining LPC model with RLS-LMS. RLS-LMS predictors are used in place of the LPC model in this model, which eliminates the predictor coefficient from the coded stream. It has rapid speed of decoding. But, Instability in numbers accompanied by a white or lightly variable signal. MPEG-4 SLS model introduced in 2006 with improved compression rate around 50% [10]. With an extra "lossless" computational layer, this approach expands on the MPEG-4 AAC lossy compression. Enhanced Scalable-to-Lossless (SLS) released in 2010 with faster decoding and encoding speed but having lower compression rate. Laplacian distribution input data is replaced with a Gaussian distribution in this model for the BPCG Entropy Block. Another adaptive coding based lossless audio codec; Enhanced Code Excited Linear Prediction (CELP) was introduced in 2010. From the processing speed perspective, it is faster than MPEG – 4 ALS but required more storage [11]. In order to eliminate intersample correlation, this model uses code-exited sample-by-sample adaptable coding. In 2013, entropy encoding based IEEE 1857.2 was introduced which have more than 50% compression rate, but processing performance is slow [12] because of Arithmetic coding's average computational complexity. In the same year another codec Sparse Linear Predictor [31] was introduced. This codec uses sparse predictors in place of LPC predictors. Although the compression ratio was higher, the decoding speed was slower with this model. In the next year, OLS and LMS filter based cascaded OLSNLMS was introduced which possess reduced computational complexities. Another popular lossless audio encoder Free Lossless Audio Codec (FLAC) released its latest version on 23rd June 2023 using MD5 and prediction model [13]. FLAC possesses around 70% average compression ratio [14]. Wavpack [15] released its latest version (5.6.0) on November 23, 2022 which have around 40% average compression ratio [14]. Famous lossless audio encoder Monkey's, which is based on integer discrete flow, achieves around 60% compression ratio [16]. An integrated model [17] of wavelet transform and Huffman encoding based lossless audio encoder introduced in 2020. In 2017 [18], a neural network-based model was introduced in which raw audio input transformed into features and processed using CNN subsequently. Lossless audio encoder based on dynamic cluster quantization technique [4] introduced In 2020. In the proposed clustering-based technique, dynamically cluster selection and bit selection for setting up the quantization level performed. In 2020, a deep learning based audio codec [5] was introduced which was based on hidden layerwise sampled value reduction technique. This technique improves the compression ratio(%) above 70% but computational complexities also increased. A model for audio compression based on deep neural networks [19] was presented in 2021 utilizing the RNN approach. As a reparametrization technique for discrete data representations, it applies the Bernoulli distribution and uses an end-to-end learning technique. This technique acheives average Signal to Distortion Ratio (SDR) of 20.53 dB with a compression ratio(%) exceeding 70%.

A Machine learning based toolkit [20] introduced recently which is used for unsupervised learning from acoustic data. The approach is based on repetitive sequential autoencoder approach which learn from time series type data using temporal motion. In these models, on the input sequence VAE is applied and RNN is applied on the output distribution subsequently to recognize the signal. Another audio compression approach that was unveiled in 2022 is the linear predictive neural net encoder (LINNE) [21]. It compresses audio by more than 60%. Also, another lossless encoding methodology based on optimum graph encoding was released in 2022 [3], and it significantly improves processing speed and compression efficiency. In 2022, The natural gradient sign algorithm (NGSA) and normalized NGSA are two adaptive algorithms that serve as the foundation for a lossless audio codec that is called NARU [32], or natural-gradient autoregressive lossless audio compressor. The utilization of a natural gradient in this work improves the sign algorithm's (SA) convergence performance. These methods significantly speed up decoding by using multiply-add operations to determine the natural gradient at each step, assuming a p-th order autoregressive model for the input data. Nonetheless, this method achieves a compression performance of about 60%.

Even though various neural network based as well as classical lossless audio compression approaches introduced, achieving compression rates like MP3 remain a mile away. Therefore, developing a lossless audio compression technique with higher compression rate and lower processing time is the need of the hour. The proposed lossless audio encoder based on neural network that achieves a higher compression ratio to fulfil this goal.

3 The technique

The proposed audio codec consists of 1D wavelet decomposition, hierarchical binary encoding and convolutional neural network enabled compressed latent space representation technique. Figure 1 displays the 1D three level wavelet decomposition tree. cA represents the approximation coefficients and cD represents the detail coefficients. Figure 3 shows the hierarchy structure.

The novelty of the proposed work lies in the integration of wavelet transformation with hierarchical encoding within a Convolutional Neural Network (CNN) architecture. This combination allows for more efficient feature extraction and data representation, leading to a significant improvement in the compression ratio. Wavelet transformation is leveraged to capture both time and frequency domain features, which are critical for effective audio compression. The hierarchical encoding further refines the data representation by breaking down the audio signal into progressively finer details, which are then processed by the CNN to identify and compress redundant information. The proposed model has been rigorously tested against both machine learning-based techniques and state-of-the-art traditional lossless audio benchmarks. The experimental results demonstrate that our model not only achieves a higher compression ratio but also maintains superior audio quality, thereby outperforming existing lossless audio techniques.

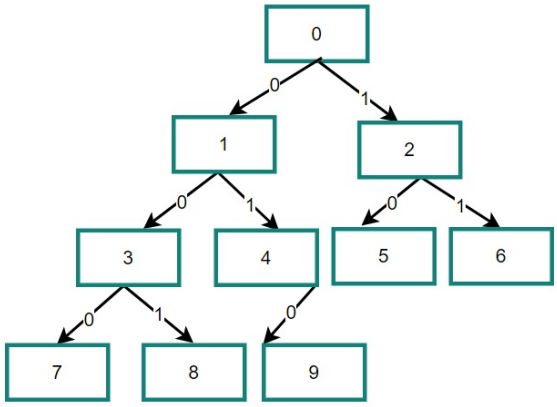


Figure 3: Hierarchical structure

Table 2 shows the nodes in each of the layer of the structure. Also, it contains the corresponding binary values of the level.

Table 2: Hierarchy levels and corresponding nodes

Level	Binary value	Nodes in the level
0	00	0
1	01	1,2
2	10	3,4,5,6
3	11	7,8,9

Algorithm 1: Encoding algorithm
Input: A slice of audio signal
Output: Compressed latent space representation of the input audio
Method: The steps are given as below

1 (Sampling): The input stream is sampled, and the sampled values are produced using the sampling process. Let f_{max} is the highest signal frequency and f_{sig} be the frequency used for sampling. The Nyquist theorem requires that the following criteria (1) be followed.

$$f_{sig} > 2x f_{max} \tag{1}$$

2 (Wavelet transformation): One dimensional three level wavelet decomposition is utilized for the input vector (created using audio sampled data) to create approximation coefficients (cA) and detail coefficients (cD). Equation (2)

represents the wavelet transformation over input audio x(t).

$$X_{a,b} = 2^{-\frac{a}{b}} \int_{-\infty}^{\infty} x(t)\Psi_{a,b}(t)dt \tag{2}$$

a and b work as frequency parameter and time respectively. $\Psi_{a,b}(t)$ demonstrates shifted and dilated variety of the mother wavelet $\Psi(t)$. It is shown in equation (3).

$$\Psi_{a,b}(t) = 2^{-\frac{a}{b}}\Psi(2^{-a}t - b) \tag{3}$$

$2^{-\frac{a}{2}}$ is constant. For wavelet transformation, which is iterative in nature, decimal valued α_{ab} coefficient is applied. A_{ab} is represented as approximation coefficient and α_{ab} as wavelet coefficient which are shown in equation 4 and 5 respectively .

$$A_{a,b}(t) = \sum_i l_{(2b-i)}A_{(a-1)i} \tag{4}$$

$$\alpha_{a,b}(t) = \sum_i h_{(2b-i)}\alpha_{(a-1)i} \tag{5}$$

The multilevel discrete 1-D wavelet transformation creates approximation coefficients (cA), detail coefficients (cD) in each level.

Step 3 (Wavelet coefficient segregation): If the coefficient is positive, add two binary digits 00 else 10 for negative numbers. Multiply the number with 10000. And get the absolute value. If the absolute value is single digit, prepend 3 zeros. If the absolute value is two digits, prepend 2 zeros and prepend 1 zero if the absolute value is three digits. Pass these 4 digits to hierarchical encoding module.

Step 4 (Hierarchical encoding):

i. Each of the digit from 0 to 9 is encoded using the hierarchical encoding technique following the figure 3 with pattern like <hierarchical level><accumulated path value> except 0.

Intput digit will be searched with the root node i.e., 0 which is on the first level. If it is matched, encode it with level number 0 and binary stream is 00. As 0 is the root node, to encode 0, no extra bit is added with level. Therefore, 0 will be coded as 00.

ii. If the input is 1 or 2, go to the table 2, corresponding level is 1 and binary stream 01. Start searching from left to right in the level 1. If the input digit is 1, it will be encoded as 010 as it is in the left node of 0. If 2, then it will be encoded as 011 as 2 is the right node of 0.

iii. If the input digits is 3 or 4 or 5 or 6, then as per the table 2, level is 2 and binary as 10. Searching start from left to right in the level 2. If the input is 3, it will be encoded as 1000 as it is in the left child of node 1. If the input is 4, it will be encoded as 1001 as it is in the right child of node 1. If the input is 5, it will be encoded as 1010 as it is in the left child of node 2. If the input is 6, it will be encoded as 1011 as it is in the right child of node 2.

iv. If the input digits is 7 or 8 or 9, then as per the table 2, corresponding level is 3 and binary as 11. Searching start from left to right in the level 3. If the input digit is 8, it will be encoded as 11001 as it is in the right child of node 3. If the input digit is 9, it will be encoded as 11010 as it is in the left child of node 9.

Step 4 (CNN encoder): These binary streams are sent to the CNN encoder for further compression.

Algorithm 2: Decoding algorithm
Input: CNN regenrated bit stream.
Output: Reconstructed original input audio.
Method: The steps are given as below

1. **Step 1 (Hierarchical decoding)**:
i. CNN reconstructed bit streams are checked. First 2 binary bits are checked, if these bits are 0, then it is the root node 0 and no extra bits need to check. It decoded as integer 0.
ii. Next 2 bits are checked, if it is 01, check the next single bit. If it is 0, decoded as integer 1, else 2.
iii. Check the next 2 bits, if it is 10, test the next 2 bits. If next 2 bits are 00, decode the integer as 3, 01 decoded as 4, 10 as 5, and 11 as 6.
iv. Test next 2 bits. If it is 11 then check next 3 bits. If next 3 bits are 000 then decoded as integer 7. If 001 then 8. If 010 then 9.
2. **Step 2 (Inverse wavelet transformation)**: Inverse wavelet is applied to reconstruct the input signal. Equation (5) is used for discrete inverse wavelet transformation.

$$x(t) = \sum_a \sum_b A_{a,b}\Psi_{a,b}(t) \tag{6}$$

3.1 CNN encoder decoder

Convolution Neural Network (CNN) [22] is very useful to extract spatial features from the dataset by using CNN kernel. In 2D CNN, Kernel slides along the two dimension. This 2D CNN is very useful in extracting features from image patterns. In the proposed technique, 2D convolutional layer is being used for feature extraction from 68 x 68 x1 input image patterns.

3.1.1 Data preprocessing

Binary data stream generated from hierarchical encodeing module preprocessed and fed as input to the CNN model. Whole binary stream is segregated into each row of 4624 columns and stored as .csv file. Each row of 4624 binary stream (0/1) is converted into into 68 x68 x 1 size image patterns sequentially. Each of the binary bit is transformed as pixel in the image. The CNN model is trained and tested with these image patterns.Figure 4 shows how the the binary strem is converted into image pattern required for CNN codec input.

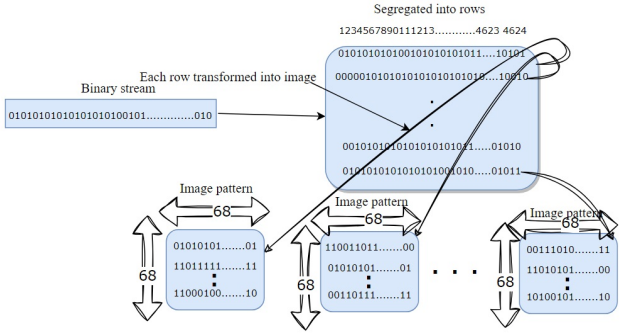


Figure 4: Image pattern formation

Table 3: CNN Model parameter configuration

Sl. no	Model Parameters	Value
1	Number of convolution layer	4
2	Convolution layer kernel size (4 layer)	(3,3)
3	Convolution layer stride	(2,2)
4	Convolution layer activation function	ReLU
5	Number of hidden units in LSTM layer	24
6	Batch size	128
7	Learning rate	0.001
8	Optimizer	Adam.
9	Loss function	MSE
10	Epochs	100

3.1.2 CNN model configuration

The proposed CNN model's computational efficiency is increased by the optimal number of setup parameters, such as convolutional layers, kernel size, stride, learning rate, and optimizer function. A trial-and-error approach is used to select the optimal parameters. Determining the optimal parameter selection is aided by observing the increased compression ratio, compression speed, and lower MSE. To compute model losses, the mean square error(MSE) is employed. In addition, the audio quality of the regenerated signal was evaluated using additional quality parameters such as PSNR, entropy, SDR, NCC, and MAE. Optimal design configuration parameters are described in table 3. The proposed CNN encoder decoder model consists of 4 convolution and 4 deconvolution layers. To extract the feature from the raw audio signal, we used 2D convolutional layer. To reconstruct the signal, deconvolutional feature extractor has been used. Adam optimization has been applied to fine tune the network. Various combinations of kernel and stride sizes were tried along with filters during the experiment. Trial and error methods were used to finalize the parameters until the least amount of loss was achieved. Input layer of the CNN accepts 68 x 68 x 1 size image generated from binary data as input, the output layer of the CNN extracts features from input, and hidden layers are used for processing purposes. With this architecture, the learning of the neural network performed for every input a weight that demonstrates a particular output. Convolution kernels are (3,3) in each layer with an activation function (ReLU). The CNN layer selected above for the experiment by trial-and-error method for getting the best result. To prevent overfitting dropout is considered as 0.1. 100 epochs were selected for evaluating the performance of the current model. Figure 11 shows the shpae and parameters of the CNN encoder model. Figure 12 shows the shpae and parameters of the CNN decoder model. Figure 13 shows the shpae and parameters of the CNN encoder decoder combined model.

3.1.3 Training

The entire CNN network is built and trained using the Tensorflow / Keras framework [15]. The ability to create network layers and train the network in according to the suggested specifications is greatly facilitated by Tensorflow / Keras. As a result, the network is trained after it converges, and appreciable decrease in training loss is seen. The final step is to evaluate, examine, and compare the total results to the established benchmark results.

The practical aspects of implementing the proposed codec are addressed by highlighting that the method can be tailored to specific needs in the audio compression field. Implementing the suggested model will lead to reduced complexity, improved compression ratio, and faster processing speeds. Additionally, it will decrease memory usage and network bandwidth requirements, making it more efficient for real-world applications

4 Experimental setup

This section discusses a number of necessary elements for the experiment, such as the environment setup, dataset preparation, data preprocessing, tools and software used, etc.

4.1 Environment

Tensorflow and Keras framework with Python 3.6 was used to implement the proposed model. Intel Core i7-4790S Processor, 16 GB RAM, 64-bit operating system, and 1 TB Hard drive were used to carry out the experiment.

4.2 Datasets

We prepared a customized dataset WLCLAC_model_training_dataset to train the WLCLAC model. Three hundred audio files, all of the same duration approximate (3 seconds), make up the dataset. Rabindra Sangeet, classical, rock, pop, and sufi are among the genres of audio songs. The training dataset was not divided into separate sets for testing. Rather, we employ 25 distinct audio tracks that fall into the five categories listed above, each lasting approximately ten seconds. The training and testing datasets are prepared using audio songs of the.wav file type, with a sampling rate of 44100 Hz. To record the

Table 4: Recording parameters

Recording parameter name	Values
Recorded file format	.wav
Recording time(training) approx.	3 seconds
Sampling rate	44100 Hz
Recording time(testing) approx.	10 seconds
Bit depth	16 bits
Channel	Mono (1)

music in.wav format and play the audio on the computer, Audacity software (Audacity 2.3.2) was utilized. The two-channel stereo audio signals are transformed to mono. The audio track has 16 bits of bit depth.

4.3 Details of recording parameters

The preparation process and parameters for the WLCLAC_audio_training_and WLCLAC_audio_testing_datasets is covered in detail in this section. Songs are recorded in.wav format using Audacity, resulting in a data collection with a 44100 Hz sample rate. The songs that comprise the customized dataset are recorded using five standard parameters. Table 4 displays the parameters' setup values.

4.4 Evaluation metrics

Evaluation of the proposed model's correctness and performance is required. In order to assess performance, several metrics are employed which are discussed below.

– Compression. Compression measures the amount of storage space the model can spare for the data. To compute the space saving, utilize equation (7) [17]. Compression ratio is a crucial metric for assessing the suggested model's capacity for compression. Here, the recommended WLCLAC approach was used to achieve average 85.72% compression while maintaining signal quality.

$$Compression(\%) = \frac{Original–compressed}{Original} \times 100 \tag{7}$$

– Mean square error(MSE). Performance of the proposed audio codec is assessed using mean square error[23]. Here, equation (8) is utilized to determine the MSE of the present model.

$$MSE = \frac{\sum_{j=1}^{n}(x_j - x'_j)^2}{n} \tag{8}$$

Where n is the number of sample points, x_j and x'_j are the actual and reconstructed values of each data point, respectively

– Entropy. Entropy is the average amount of information contained in a symbol or variable [24]. The unpredictable nature is shown by the entropy. Equation (9) is used to calculate the entropy for the proposed model.

$$H(X) = \sum_{k=1}^{m} P(x_k)\log_2 P(x_k) \tag{9}$$

H(x) indicates entropy of x. Here, x denotes the random variable. It takes values from the set of values $x_1, x_2, ...x_m$ corresponding probabilities $P(x_1), P(x_2), ..P(x_m)$ $where$ $\sum_{k=1..m} P(x_k) = 1$

– PSNR(in dB). A technique for assessing the quality of the original signal in compressed audio files is the peak signal to noise ratio (PSNR) [25] [26]. Equation (10) is used to calculate PSNR

$$PSNR = IJ\frac{\sum_{i,j} X_{i,j}^2}{\sum_{i,j}(X_{i,j} - \overline{X}_{i,j})^2} \tag{10}$$

Here, $X_{i,j}$ denotes the original values and is represented by $\overline{X}_{i,j}$ reconstructed values.

– Normalised cross correlation. Input and reconstructed audio signals are compared using NCC. Higher correlation is indicated by a higher NCC. Two identical signals result in a score of 1. NCC is calculated using equation (11).

$$NCC = \frac{\sum_{i,j} X_{i,j}\overline{X}_{i,j}}{\sum_{i,j} X_{i,j}^2} \tag{11}$$

Where $\overline{X}$ are the reconstructed values, and X are the input values to the model.

– Mean Absolute Error(MAE). The average of the variations between the generated values and the original values is referred to as the "mean absolute error" [27]. This measure displays the variations between the input and the reconstructed value. Equation (12) gives the following illustration of it:

$$MAE = \frac{\sum_{j}^{n}|X_j - X'_j|}{n} \tag{12}$$

– Signal distortion ratio(SDR): The experiment uses SDR[28] to measure the reconstructd signal's audio quality. The SDR uses decibels (dB). SDR is a measure of how close the reconstructed signal(S_{recon}) was to the original signal(S_{orig}). The calculation is as follows. The proposed method produces an average Signal to Distortion Ratio (SDR) of 42.45 dB.

$$SDR = 10log_{10}\frac{|S_{orig}|^2}{|S_{recon} - S_{orig}|^2} \tag{13}$$

5 Results and analysis

The wavelet decomposition module receives all of the audio from the WLCLAC_audio_training_dataset. Using the proposed hierarchical binary encoder, the appropriate wavelet coefficients are encoded into binary patterns. The CNN encoder-decoder model uses all of the binary data streams that match the training dataset as input. 30% of the CNN model's input training dataset is used for validation and 70% for training. The CNN model testing dataset is generated in a similar manner from the WLCLAC_audio_testing_dataset. The WLCLAC codec is tested individually using this testing dataset. In order to compute the compression independently, the three stages of the suggested codec are assessed during the experiment. The results are thoroughly explained in Section 4.1.

5.1 Experimental results

Three existing conventional lossless audio compression techniques like Monkey's Audio [29], Wavpack Lossless [15], and FLAC [13] are considered as referenced systems to evaluate the performance of the proposed model. Table 5 shows the compression performance of the proposed codec with the three existing codecs and it is evident that the current model achieves 85.72% (shown in table 5) compression which is higher than 56.45%, 51.18%, and 70.64% compressions achieves by Monkey's Audio [29], Wavpack Lossless [15], and FLAC [13] respectively. Figure 5 shows the graphical representation of the compression, PSNR, and entropy achieved using WLCLAC with reference to existing referenced lossless audio compression techniques.

Using a variety of audio tracks, the compression speed implies encoding and decoding times of the proposed technique are compared with other state-of-the lossless audio codecs. The resultset is shown in table 6 and it is evident that a reduction in the encoding and decoding times of the suggested approach translates into an increase in compression speed. Figure 6 compares the encoding and decoding speed graphically. For all the parameters, WLCLAC achieves better results.

Table 5: Performance of the proposed model in relation to the existing audio codecs

Method	Compression(%)	Entropy	PSNR(dB)
Monkey's Audio	56.45	13.35	52.98
Wavpack	51.18	13.55	52.12
FLAC	70.64	13.56	52.56
WLCLAC	85.72	13.67	56.56

Also, we have done the robustness performance evaluation of the proposed model with 3 others existing deep learning based lossless audio codecs : i) DLLAE [5] ii) Daniela N. Rim et al.[19] iii) LINNE [30]. Proposed WLCLAC model is capable of regenerating the original audio signal with very negligible deviations. Mean square error

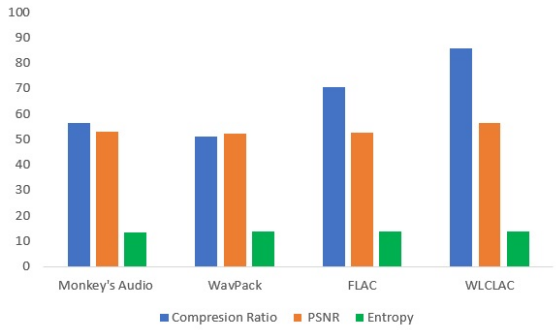


Figure 5: Comparison of the compression and quality parameters for WLCLAC

Table 6: Encoding and decoding time comparison

Method	Encoding(sec.)	Decoding(sec.)
Monkey's Audio	0.06782	0.06802
Wavpack	0.07134	0.07211
FLAC	0.06871	0.06921
WLCLAC	0.06321	0.06431

value evaluated for the proposed model is 0.001822. Also, the RMSE of the proposed model is 0.042684. MAE value calculated for the proposed model is 0.033912. Close to 0 value of the MAE and RMSE indicates the robustness of the system and close similarities between original and predicted signal. Also, another parameter called NCC used to measure the regenerated signal quality is 0.998761. Closer to 1 NCC value indicates regeneration is good. Table 7 shows the evaluated values of the parameters like MSE, RMSE, MAE, and NCC of the current model with respect to the other referenced prediction system to demonstrate the accuracy and robustness of the model. Table 8 compares the compression ratio(%) produced by the new approach to the other DNN model. According to table 8, the proposed lossless audio codec produces more compression than the existing models. Figure 7 shows the graphic comparison of category-wise compression of compression ratio (%) of the proposed model with other existing neural network based models.

Table 7: Robustness performance comparison with exiting DNN model

Method	MSE	RMSE	MAE	NCC
WLCLAC	0.001822	0.042684	0.33912	0.998761
DLLAE [5]	0.017872	0.133686	1.161451	0.985634
Daniela N. Rim et al.[19]	0.134536	2.185838	2.617224	0.981232
LINNE[30]	0.125162	0.158625	2.017224	0.986578

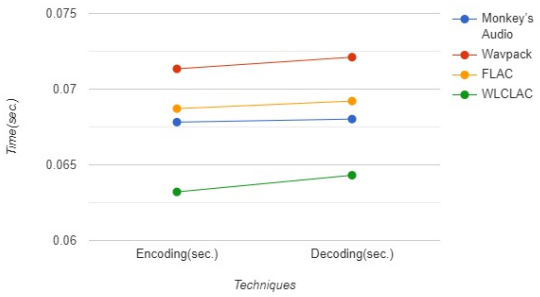


Figure 6: Comparison of the compression speed

Table 8: Categorywise compression comparison

Method	Pop	Sufi	Ghazal	Rabi	Classical
DLLAE [5]	87.18	86.47	86.01	87.01	86.12
Daniela N. Rim et al. [19]	84.21	82.34	82.15	83.81	83.27
LINNE [30]	73.76	75.32	73.38	71.13	73.48
Proposed method	85.18	85.01	86.89	85.4	86.12

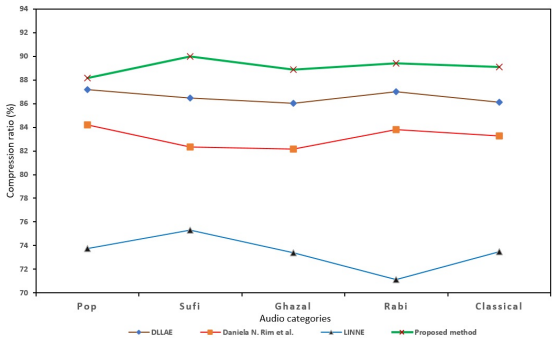


Figure 7: Comparison of the compression ratio(%)with existing DNN models

Figure 8 represents the regenerated audio signal by the WLCLAC model. Therefore, it is visible that regenerated signal is like original signal with negligible deviations. Therefore, the experimental data set demonstrates that the proposed method, when compared with existing standard audio compression approaches, acquired a higher compression ratio and improved audio regeneration quality.

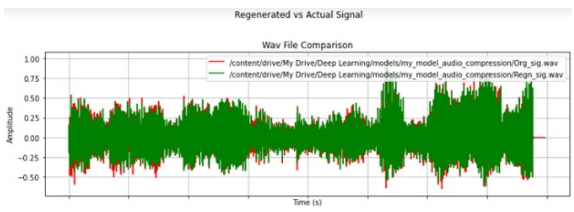


Figure 8: Comparison of the original and regenerated signal

Figure 9 shows the epochwise loss of the model. Figure 10 shows the epochwise training pregress and corresponding mse loss of the model.

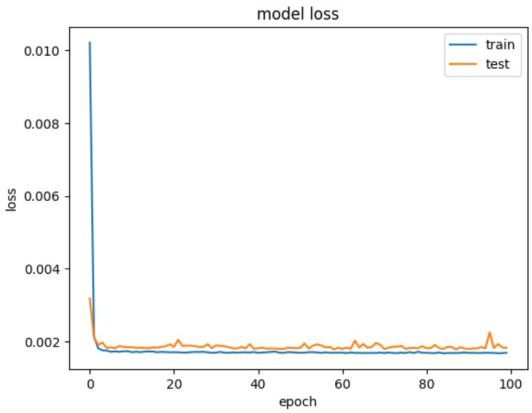


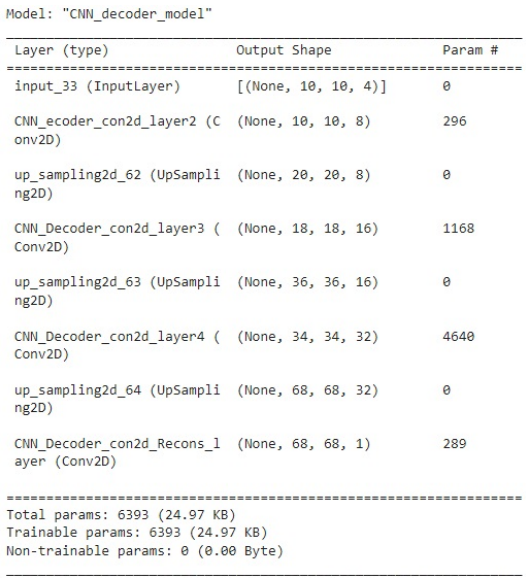
Figure 9: Model loss

```
Epoch 92/100
343/343 [==============================] - 4s 10ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 93/100
343/343 [==============================] - 4s 12ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 94/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 95/100
343/343 [==============================] - 4s 13ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 96/100
343/343 [==============================] - 5s 13ms/step - loss: 0.0017 - val_loss: 0.0023
Epoch 97/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 98/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0019
Epoch 99/100
343/343 [==============================] - 4s 13ms/step - loss: 0.0017 - val_loss: 0.0018
Epoch 100/100
343/343 [==============================] - 4s 11ms/step - loss: 0.0017 - val_loss: 0.0018
Running prediction..
85/85 [==============================] - 1s 5ms/step
Plotting results..
metrics name is ['loss']
```

Figure 10: Epochwise training progress

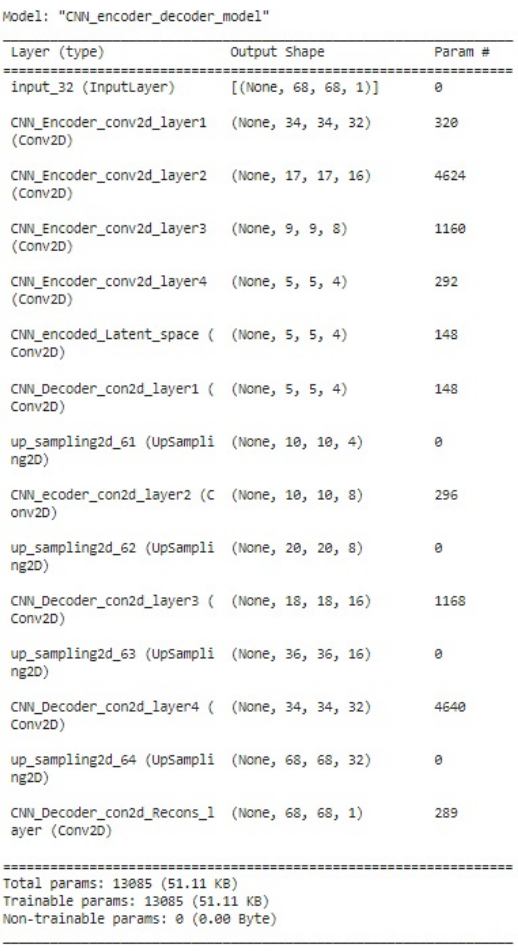
```
Model: "CNN_encoder_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_32 (InputLayer)       [(None, 68, 68, 1)]       0

 CNN_Encoder_conv2d_layer1   (None, 34, 34, 32)        320
 (Conv2D)

 CNN_Encoder_conv2d_layer2   (None, 17, 17, 16)        4624
 (Conv2D)

 CNN_Encoder_conv2d_layer3   (None, 9, 9, 8)           1160
 (Conv2D)

 CNN_Encoder_conv2d_layer4   (None, 5, 5, 4)           292
 (Conv2D)

 CNN_encoded_Latent_space (  (None, 5, 5, 4)           148
 Conv2D)

=================================================================
Total params: 6544 (25.56 KB)
Trainable params: 6544 (25.56 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 11: Shape and parameters of the CNN encoder model



```
Model: "CNN_decoder_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_33 (InputLayer)       [(None, 10, 10, 4)]       0

 CNN_ecoder_con2d_layer2 (C  (None, 10, 10, 8)         296
 onv2D)

 up_sampling2d_62 (UpSampli  (None, 20, 20, 8)         0
 ng2D)

 CNN_Decoder_con2d_layer3 (  (None, 18, 18, 16)        1168
 Conv2D)

 up_sampling2d_63 (UpSampli  (None, 36, 36, 16)        0
 ng2D)

 CNN_Decoder_con2d_layer4 (  (None, 34, 34, 32)        4640
 Conv2D)

 up_sampling2d_64 (UpSampli  (None, 68, 68, 32)        0
 ng2D)

 CNN_Decoder_con2d_Recons_l  (None, 68, 68, 1)         289
 ayer (Conv2D)

=================================================================
Total params: 6393 (24.97 KB)
Trainable params: 6393 (24.97 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 12: Shape and parameters of the CNN decoder model



```
Model: "CNN_encoder_decoder_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_32 (InputLayer)       [(None, 68, 68, 1)]       0

 CNN_Encoder_conv2d_layer1   (None, 34, 34, 32)        320
 (Conv2D)

 CNN_Encoder_conv2d_layer2   (None, 17, 17, 16)        4624
 (Conv2D)

 CNN_Encoder_conv2d_layer3   (None, 9, 9, 8)           1160
 (Conv2D)

 CNN_Encoder_conv2d_layer4   (None, 5, 5, 4)           292
 (Conv2D)

 CNN_encoded_Latent_space (  (None, 5, 5, 4)           148
 Conv2D)

 CNN_Decoder_con2d_layer1 (  (None, 5, 5, 4)           148
 Conv2D)

 up_sampling2d_61 (UpSampli  (None, 10, 10, 4)         0
 ng2D)

 CNN_ecoder_con2d_layer2 (C  (None, 10, 10, 8)         296
 onv2D)

 up_sampling2d_62 (UpSampli  (None, 20, 20, 8)         0
 ng2D)

 CNN_Decoder_con2d_layer3 (  (None, 18, 18, 16)        1168
 Conv2D)

 up_sampling2d_63 (UpSampli  (None, 36, 36, 16)        0
 ng2D)

 CNN_Decoder_con2d_layer4 (  (None, 34, 34, 32)        4640
 Conv2D)

 up_sampling2d_64 (UpSampli  (None, 68, 68, 32)        0
 ng2D)

 CNN_Decoder_con2d_Recons_l  (None, 68, 68, 1)         289
 ayer (Conv2D)

=================================================================
Total params: 13085 (51.11 KB)
Trainable params: 13085 (51.11 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 13: Shape and parameters of the CNN encoder decoder model

The Mean Opinion Score (MOS) is used to assess the perceptual quality of the regenerated audio signals . Table 9 displays the MOS measurement of the regenerated audio quality. According to table 9, a sound quality grade of ”5” indicates ”Excellent” sound, while a grade of ”1” indicates ”Bad” sound. The ITUR Rec. 500 quality rating is appropriate for the present quality measuring activities. because it offers a quality rating ranging from 1 to 5 [33]. The MOS (mean opinion score) number for the various categories evaluated by the current technique, 5, indicates that the reconstructed audio quality remains unaffected by this tiny data change throughout the transformation, since it is over the threshold level of human perception.

Table 9: A rating system to evaluate the decline in audio quality.

Rating	Impairment	Quality
1	Very annoying	Bad
2	Annoying	Poor
3	Slightly annoying	Fair
4	Perceptible, not annoying	Good
5	Imperceptible	Excellent

6 Conclusion and future scope

The proposed model has been trained and validated using real-time audio data. The proposed model’s performance is assessed in comparison with those of current standard lossless audio codecs. The mean square error of the robust model is very less. Compression of the proposed model is 85.72%, which is higher compared with existing lossless audio codecs. The computational time for the model is lower than the referenced systems. The future scope of the work is to enhance the computational performance and compression by enhancing the model.

References

[1] Nowak, N., Zabierowski, W.(2011): Methods of sound data compression–comparison of different standards. Radio electronics and informatics (4), 92–95.

[2] Sharma, K., Gupta, K.(2017), Lossless data compression techniques and their performance. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 256–261, IEEE. https://doi.org/10.1109/CCAA.2017.8229810

[3] Mondal, U.K., Debnath, A.(2022), Designing a novel lossless audio compression technique with the help of optimized graph traversal (lacogt). Multimedia Tools and Applications 81(28), 40385–40411.

[4] Mondal, U.K., Debnath, A.(2021), Developing a dynamic cluster quantization based lossless audio compression (dcqlac). Multimedia Tools and Applications 80(6), 8257–8280.

[5] Mondal, U.K., Debnath, A., et al.(2020), Deep learning-based lossless audio encoder(dllae). In: Intelligent Computing: Image Processing Based Applications, pp. 91–101. Springer. https://doi.org/10.1007/978-981-15-4288-6_6

[6] Mondal, U.K., Debnath, A., et al.(2023), Designing an iterative adaptive arithmetic coding-based lossless bio-signal compression for online patient monitoring system(iaalbc). In: Frontiers of ICT in Healthcare: Proceedings of EAIT 2022, pp. 655–664. Springer. https://doi.org/10.1007/978-981-19-5191-6_53

[7] Holighaus, N., Koliander, et al.(2019), Characterization of analytic wavelet transforms and a new phaseless reconstruction algorithm. IEEE Transactions on Signal processing 67(15), 3894–3908.

[8] Jmour, N., Zayen, S., Abdelkrim, A.(2018), Convolutional neural networks for image classification. In: 2018 International Conference on Advanced Systems and Electric Technologies (IC ASET), pp. 397–402,IEEE.

[9] Reznik, Y.A.(2004), Coding of prediction residual in mpeg-4 standard for lossless audio coding (mpeg-4 als). In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, p. 1024,IEEE. https://doi.org/10.1109/ICASSP.2004.1326722

[10] Yu, R., Lin, X., Rahardja, S., Huang, H. (2005), Mpeg-4 scalable to lossless audio coding-emerging international standard for digital audio compression. In: 2005 IEEE 7th Workshop on Multimedia Signal Processing, pp. 1–4,IEEE. https://doi.org/10.1109/MMSP.2005.248562

[11] Wei, B., Wang, J., Gibson, J.D. (2001), Enhanced celp coding with discrete spectral modeling. In: Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No. 01EX489), pp. 111–113,IEEE. https://doi.org/10.1109/ISIMP.2001.925344

[12] Gunawan, T.S., Zain, M.K.M., Muin, F.A., Kartiwi, M. (2017), Investigation of lossless audio compression using ieee 1857.2 advanced audio coding. Indonesian Journal of Electrical Engineering and Computer Science 6(2), 422–430. https://doi.org/10.11591/ijeecs.v6.i2.pp422-430

[13] Coalson, J.: Xiph. Org Foundation,"FLAC: Free lossless audio codec". https://xiph.org/flac/index.html. Accessed:15-10-2024.

[14] Tu, W., Yang, Y., Du, B., Yang, W., Zhang, X., Zheng, J.(2020), Rnn-based signal 339 classification for hybrid audio data compression. Computing 102(3), 813–827. https://doi.org/10.1007/s00607-019-00713-8

[15] http://www.wavpack.com/. Accessed: 15-10-2024.

[16] Oquab, M., Bottou, L., Laptev, I., Sivic, J.(2015), Is object localization for free?-weakly supervised learning with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 685–694. https://doi.org/10.1109/CVPR.2015.7298668

[17] Debnath, A., Mondal, U.K., et al. (2020), Achieving lossless audio encoder through integrated approaches of wavelet transform, quantization and huffman encoding (laeiwqh). In: 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1–5, IEEE. https://doi.org/10.1109/ICCSEA49143.2020.9132865

[18] Zeng, Ming, and Huahong Zeng. "Research on Violin Audio Feature Recognition Based on Mel-frequency Cepstral Coefficient-based Feature Parameter Extraction." Informatica 48, no. 19 (2024).

[19] Rim, D.N., Jang, I., Choi, H.(2021) Deep neural networks and end-to-end learning for audio compression. arXiv preprint arXiv:2105.11681. https://doi.org/10.5626/JOK.2021.48.8.940

[20] Freitag, M., Amiriparian, S., et al. (2017), audeep: Unsupervised learning of representations from audio with deep recurrent neural networks. The Journal of Machine Learning Research 18(1), 6340–6344. https://doi.org/10.48550/arXiv.1712.04382

[21] Mineo, T., Shouno, H.: A lossless audio codec based on hierarchical residual prediction. (2022), In: 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 123–130, IEEE. https://doi.org/10.23919/APSIPAASC55919.2022.9980327

[22] Kadhim, A.R., Khudeyer, R.S. and Alabbas, M., 2024. Facial Sentiment Analysis Using Convolutional Neural Network and Fuzzy Systems. Informatica, 48(12).

[23] Wang, K., Qi, X., Liu, H. (2019), Photovoltaic power forecasting based lstm convolutional network. Energy 189, 116225. https://doi.org/10.1016/j.energy.2019.116225

[24] Shannon, C.E. (1948), A mathematical theory of communication. The Bell system 366 technical journal 27(3), 379–423.

[25] Kutter, M., Petitcolas, F.A.: Fair benchmark for image watermarking systems.(1999),In: Security and Watermarking of Multimedia Contents, vol. 3657, pp. 226–239 International Society for Optics and Photonics. https://doi.org/10.1117/12.344672

[26] Manju, M., Abarna, P., Akila, U., Yamini, S.(2018),Peak signal to noise ratio & mean square error calculation for various image patterns using the lossless image compression in ccsds algorithm. International Journal of Pure and Applied Mathematics 119(12),14471–14477.

[27] Jie Y. Design and Application of Neural Network-based Bp Algorithm in Speech Translation Robot. Informatica. 2023 Aug 4;47(7).

[28] Krizhevsky, A., Sutskever, I., Hinton, G.E.(2012), Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25. https://doi.org/10.1145/3065386

[29] https://monkeysaudio.com/index.html. Accessed: 15-10-2024.

[30] Mineo, T., Shouno, H.(2022), A lossless audio codec based on hierarchical residual prediction. In: 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 123–130, IEEE. https://doi.org/10.23919/APSIPAASC55919.2022.9980327

[31] Giacobello D, Christensen MG, Murthi MN, Jensen SH, Moonen M. Sparse linear prediction and its applications to speech processing. IEEE Transactions on Audio, Speech, and Language Processing. 2012 Feb 3;20(5):1644-57. https://doi.org/10.1109/TASL.2012.21868071

[32] Mineo, T., Shouno, H.: Improving sign-algorithm convergence rate using natural gradient for lossless audio compression. EURASIP Journal on Audio, Speech, and Music Processing 2022(1), 12 (2022). https://doi.org/10.1186/s13636-022-00243-w

[33] Arnold M (2000) Audio watermarking: features, applications and algorithms. In: 2000 IEEE International conference on multimedia and expo. ICME2000. Proceedings. Latest advances in the fast changing world of multimedia (cat. no. 00TH8532), vol 2. IEEE, pp 1013–1016. https://doi.org/10.1109/ICME.2000.871531

Classification and Identification of Weeds Using Machine Learning Classifiers

V. Padma Sree, Akhila John Davuluri*, Siddaiah Polepalli
Department of Electronics and communication engineering, Acharya Nagarjuna University, Guntur-522510, Andhra Pradesh, India
E-mail: padmasree4396@gmail.com, akhilajohndavuluri@gmail.com, siddaiah_p@yahoo.com
*Corresponding author

Keywords: cnn, uav, machine learning algorithms

Received: September 2, 2023

Weeds pose significant challenges in agriculture, impacting crop yields and increasing the reliance on herbicides. Accurate and timely identification of weeds is crucial for effective weed management strategies. This study proposed a novel approach for automated identification of weeds using various machine learning classifiers. Our study explores the effectiveness of diverse algorithms, including Support Vector Machine (SVM), Random Forest, Decision Tree, k-Nearest Neighbors (KNN), Extra Tree, and Gaussian Naive Bayes (NB). By pre-processing and engineering features from a diverse dataset of weed images, we ensure optimal model performance. Through rigorous experimentation and evaluation, we assess the performance of each classifier in weed identification. Notably, the Extra Tree classifier achieves an impressive accuracy of 96.35% and an outstanding kappa coefficient of 96.21%. These findings offer valuable insights into the effectiveness of different classifiers and their potential applications in precision agriculture for targeted weed management and crop optimization

Povzetek: Analizirana je uporaba strojnega učenja za avtomatizirano prepoznavanje plevela v kmetijstvu, vključno s SVM, naključnimi gozdovi in CNN, ter uporaba UAV slik.

1 Introduction

Weed identification plays a pivotal role in agriculture and environmental management, involving the distinction of unwanted plant species from desired vegetation. The conventional manual process is laborious and time-intensive, prompting the integration of machine learning and computer vision techniques for automated identification. Utilizing tools like Convolution Neural Networks (CNNs) and ensemble classifiers, modern approaches analyze visual features, leaf shapes, and textures in captured images to efficiently detect and classify weeds. This technological advancement enhances accuracy and expedites the identification process, with applications extending to agriculture, ecological conservation, and land management for optimized resource utilization and sustainable practices [1]. In a study by [2], the utilization of a Random Forest Classifier for weed identification yielded an initial accuracy of 82% and a kappa coefficient of 0.73 in preliminary assessments. A study [3] used tiny YOLOv3 for Convolvulus sepium detection in sugar beet fields. They combined 2271 synthetic images with 452 field images for model training. YOLO anchor box sizes were determined via k-means clustering on the training dataset. Testing on 100 field images showed that using the combination of synthetic and original images provided improved mAP from 0.751 to 0.829 compared to using field images alone. In a study detailed in [4], the differentiation of crops and weeds based on visible and near-infrared spectrums is achieved through the application of Support Vector Machine, Artificial Neural Network, and Decision Tree techniques. The research attains a notable accuracy of 68.40%. The investigation outlined in [5] delves into an automated weed detection system that employs Convolutional Neural Networks (CNN) with Unmanned Aerial Vehicle (UAV) imagery. The proposed CNN LVQ (Learning Vector Quantization) model emerges as a remarkable contender for effectively classifying various categories. Notably, the soil class achieves an impeccable 100% user accuracy, closely trailed by soybean (99.79%), grass (98.58%), and broadleaf (98.32%). After meticulous hyper parameter refinement, the developed CNN LVQ model achieves an exceptional overall accuracy of 99.44% for weed detection, decisively surpassing the performance of previously documented studies. Within the domain of machine learning, a multitude of techniques are harnessed for weed identification. An illustrative instance involves the application of machine learning methodologies for weed detection in an Australian chilli crop field. In this context, diverse algorithms, including random forest (RF), support vector machine (SVM), and k-nearest neighbours (KNN), are systematically explored to ascertain their efficacy in leveraging UAV images for weed detection. The achieved results underscore notable accuracies: 96% for RF, 94% for SVM, and 63% for KNN, as documented in reference [6]. In a recent scholarly investigation [7], the identification of weeds within vegetable plantations was accomplished employing Centre Net, a fusion of deep learning and image proce

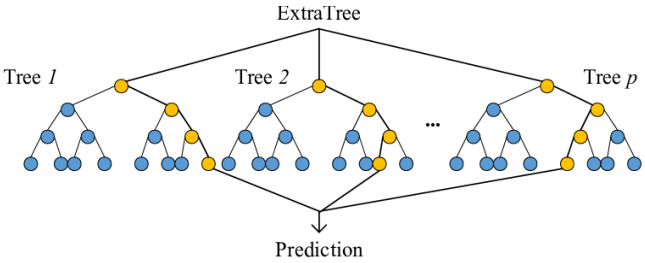


Figure 1: Extra tree classifier

Remarkable outcomes, with a precision of 95.6%, a recall of 95.0%, and an impressive F1 score of 0.953. To further enhance Bayesian classification accuracy, Genetic Algorithms (GAs) were skilfully employed to optimize the colour index. A hybrid CNN-SVM classifier is proposed for weed recognition in winter rape fields, aiming to improve accuracy. Utilizing VGG network model, the approach achieved average accuracies of 99.2% in training and 92.1% in classification [8]. This [9] overview explores RNN- and CNN-based weed detection within crop enhancement, showcasing deep learning's potential for agriculture challenges. The Convolutional Neural Network emerges as the most Efficient technique for weed detection, leading to the development of a smart system for in-place weed identification and spraying. In study [10], the SLIC-RF algorithm is proposed for differentiating crops and weeds in upland rice fields using UAV imagery, achieving accuracies up to 0.915. The approach combines HSV-based SLIC with various features, demonstrating potential for effective site-specific weed management. In their study [11] proposed an automatic weed mapping method using UAV imagery in oat fields. Four classification algorithms were tested, with the automatic object-based approach achieving the highest accuracy of 89.0% and 87.1% for two subsets, enabling potential use in precision weed treatment. In their work [12] developed a vision-based weed detection system for soybean crops using custom lightweight deep learning models. Their proposed 5-layer CNN architecture achieved a high accuracy of 97.7% with minimal latency and memory usage, promising efficiency and productivity enhancements in the soybean industry. In their work [13] introduced a method utilizing deep features and one-class classification on unsupervised data for weed detection in UAV images. The approach achieved up to 90% accuracy on test datasets, comparable to supervised models, by employing one-class classifier trained on crop row-detected unsupervised data. This research [14] assesses the Random Forest (RF) classifier's effectiveness in classifying forest cover from Land sat TM imagery, achieving a 96% accuracy using auxiliary data and systematically collected aerial photography. The study presents an operational and cost-effective approach for generating accurate forest cover maps across diverse sclerophyll forests using open-source software. In [15] The realm of smart agriculture, a revolutionary robotic system diminishes reliance on traditional spraying methods like pesticides and herbicides, aiming to meet global food demands and enhance crop production. To achieve this, a Deep Learning (DL) approach utilizing a blend of Convolutional Neural Networks (CNN) and Long-Short-Term Memory (LSTM) is proposed for weed identification and classification. The method achieves an impressive 99.36% average classification accuracy across nine weed categories, surpassing other established techniques. A study [16] employed YOLOv3 deep learning to identify volunteer cotton plants amidst corn fields using UAV-captured RGB images, achieving over 80% detection accuracy and highlighting the potential of DL for real-time pest mitigation via computer vision and UAV technology.

In this context, this study introduces a hybrid methodology that integrates Convolutional Neural Networks (CNN) with the Extra Trees classifier to address the task of identifying weeds within agricultural fields. The CNN part specializes in capturing distinguishing features from input images, and the Extra Trees classifier employs these acquired features to execute the classification process. By merging the advantages of deep learning and the Extra Trees approach, this novel method strives to achieve both accurate and efficient weed identification outcomes.

Leveraging advanced technologies such as Convolutional Neural Networks (CNNs) and the Extra Trees classifier offers a promising avenue for automating weed identification in agriculture. This approach, combining CNNs and Extra Trees, has the potential to enhance accuracy, streamline efficiency, and optimize resource utilization. Consequently, it holds the key to transforming weed management practices and driving improved agricultural productivity.

This study presents an innovative approach that merges Convolutional Neural Networks (CNN) for feature extraction and classification with the Extra Trees classifier to identify weeds in diverse crops, with a specific focus on soybean. Additionally, it differentiates between grass and broadleaf weeds. The dataset encompasses 4400 UAV images, spanning categories like soybean, soil, grass and broadleaf. Notably, the method enhances the performance of the Extra Trees classifier.

This optimized setup improves training effectiveness and maximizes the utility of the extensive UAV weed dataset. The subsequent sections of the paper are structured as follows: Section 2 introduces the CNN model, while Section 3 describes various machine learning algorithms, followed by Section 4 which elaborates the dataset used, and Section 5 presents the methodology employed in this paper. Section 6, presents the experimental results. Subsequently, Section 7 discusses the results and discussion and the conclusion is provided in Section 8.

2 Convolutional neural network

This research leverages the capabilities of Convolutional Neural Networks (CNNs) to serve as a comprehensive framework for both feature extraction and classification. CNNs have garnered substantial recognition for their prowess in processing grid-like data, particularly images, and have exhibited exceptional performance in critical computer vision tasks such as object detection and image classification. In this study, a sophisticated CNN architecture is employed, comprising several pivotal layers, namely convolutional layers, pooling layers, Global Average Pooling (GAP) layers, and fully Connected layers. The architecture begins with the implementation of multiple Convolutional layers, strategically employing learnable filters to extract essential features at distinct spatial resolutions. Three Convolutional layers are thoughtfully configured, with each layer tailored to specific characteristics. Subsequently, pooling layers are introduced to the design, tasked with spatially down-sampling the feature maps while preserving vital information that underpins accurate classification. Of notable importance is the incorporation of the Global Average Pooling (GAP) layer, a fundamental component that furthers feature consolidation. By performing average pooling across the entirety of the feature map, the GAP layer efficiently condenses spatial information while retaining the most critical features. This contributes to the network's resilience against translation variances, thereby enhancing its ability to generalize.

.

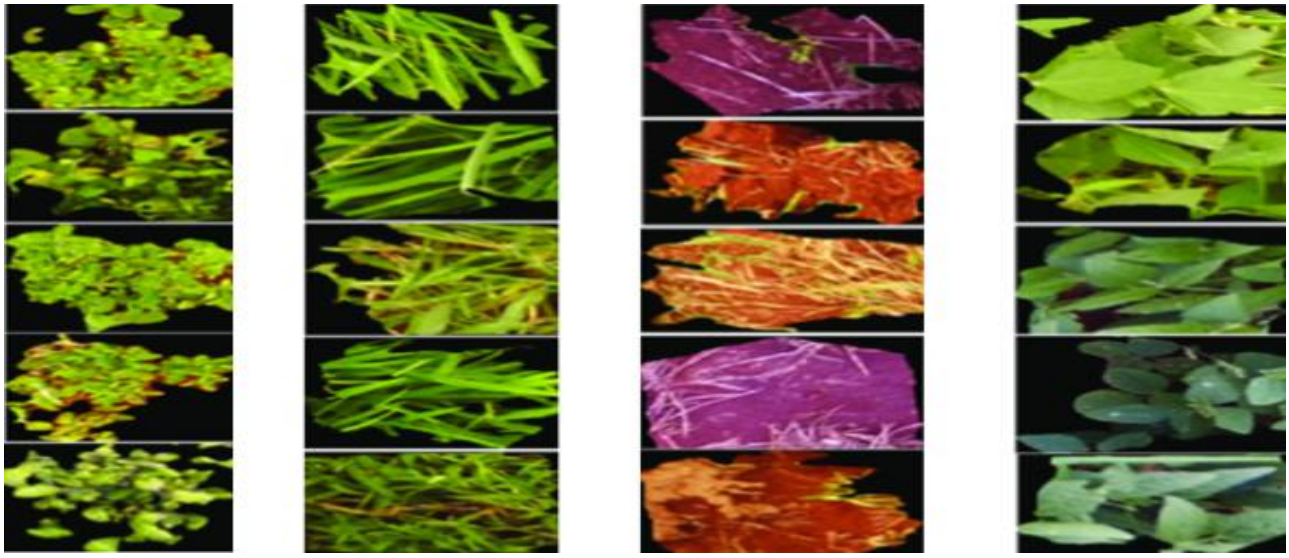


Figure 2: Raw UAV Images of Broadleaf, Grass, Soil, Soya bean

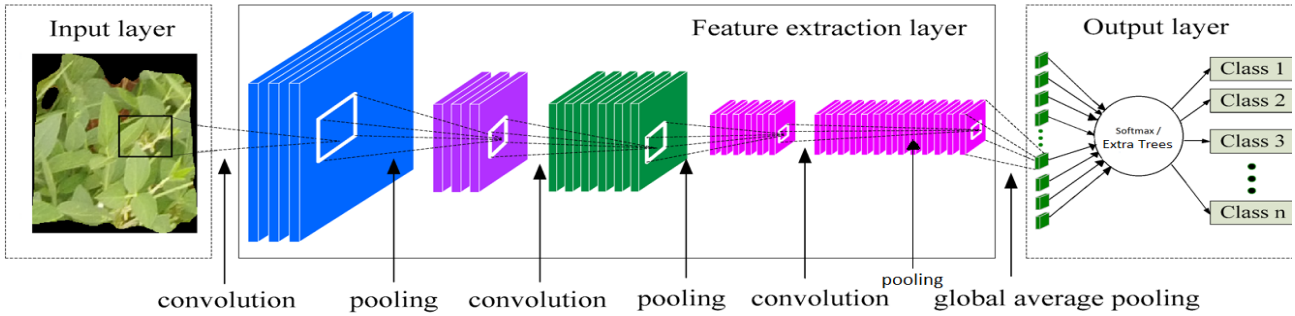


Figure 3: Architecture of proposed model

Building on these foundations, the architecture integrates fully connected layers, which serve to aggregate the extracted features and drive accurate predictions. Through the application of softmax activation, these fully connected layers translate the feature amalgamation into class probabilities, particularly beneficial for multi-class classification tasks.

This unified framework embodies the remarkable Potential of CNNs in deciphering complex features and facilitating precise classification, thereby serving as a cornerstone in advancing state-of-the-art image analysis methodologies.

3 Machine learning algorithms used

Machine learning algorithms are computational models designed to enable computers to learn from data and improve their performance on specific tasks without

being explicitly programmed. They fall into categories such as supervised learning, where algorithms are trained on labeled data to make predictions, unsupervised learning, where they find patterns in unlabeled data, semi-supervised learning, combining elements of both supervised and unsupervised learning, reinforcement learning, where agents learn to take actions to maximize rewards, and deep learning, which involves training deep neural networks. The choice of algorithm depends on the problem at hand, available data, and desired outcomes. The successful application of machine learning has transformed various industries, allowing computers to make complex decisions and predictions once exclusive to human intelligence.

3.1 SVM

Support Vector Machines (SVMs) are classification algorithms that seek the best hyper plane to separate different classes by maximizing the margin. They focus on support vectors, near the decision boundary. SVMs handle high dimensions, outliers, and complex data via kernels. While binary by design, they extend to multi-class scenarios using strategies like One-vs-Rest or One-vs-One.

3.2 Random forest

The Random Forest classifier is an ensemble learning algorithm that combines multiple decision trees, each trained on random subsets of data and features. This aggregation of predictions improves accuracy, reduces over fitting, and offers insights into feature importance, making it a robust and popular choice for classification and regression tasks.

3.3 Decision tree

A Decision Tree is a machine learning algorithm that uses a tree-like structure to make decisions based on features. Each internal node represents a decision, branches indicate outcomes, and leaf nodes give predictions. While intuitive and suitable for non-linear data, they can over fit and struggle with generalization.

3.4 Gaussian naive bayes

Gaussian Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that features follow a Gaussian distribution and are conditionally independent given the class label. It's particularly useful for continuous numerical data and works well even with limited training data. Despite its simplicity and the naive assumption, it often performs surprisingly well in real-world scenarios and is a popular choice for text and image classification tasks.

3.5 KNN

K-Nearest Neighbours (KNN) is a supervised algorithm for classification and regression. It identifies the K closest training data points to a query point and predicts Using the majority class or average value of neighbors. KNN assumes similar data points share outcomes, offering simplicity but sensitivity to distance metric and K value.

3.6 Extra tree

The Extra Trees classifier, short for Extremely Randomized Trees classifier, is an ensemble learning algorithm used for classification tasks. It builds multiple decision trees using random subsets of features and data. Unlike Random Forest, Extra Trees selects random splits at each node, reducing variance and enhancing generalization. This randomness also makes it computationally efficient. The algorithm aggregates predictions from all trees to make the final classification decision, resulting in improved accuracy and reduced over fitting. Extra Trees is shown in Figure 1.

4 Dataset

This study employs a dataset sourced from [21], consisting of 400 UAV snapshots of soybean crops captured from a 4 m altitude using the DJI Phantom 3 Professional. The images have a ground sampling distance of 1 cm, as shown in Fig. 2. The images underwent segmentation using the SLIC algorithm, resulting in 15,336 segments in the dataset. Of these segments, 7,376 correspond to soybean, 3,520 to grass, 3,249 to soil, and 1,191 to broadleaf weeds. For more comprehensive dataset information, refer [21]. The dataset is used for weed identification in soybean crops, as well as distinguishing the crops from grass, soil, and broadleaf weed classes. Using classifier ensemble approach rice crop yield is predicted in India [22].

5 Methodology

In this study, 15,334 images were randomly selected from a total collection of 15,336 photographs within the dataset. The dataset encompasses four classes: broadleaf, grass, soil, and soybean. The dataset was divided into a ratio of 80:20 for training and testing. The images were processed, and a CNN with 12 layers was developed for imagery classification (Fig. 3). The model was built using Keras 2.3.0 API with Tensor Flow 2.0 backend and Python 3.8. It consists of three hidden convolution layers, three max-pooling layers, a Global Average Pooling (GAP) layer, and dense layers. The Convolutional layers employ the Rectified Linear Unit (ReLU) activation function to capture complex patterns through non-linearity. Max-pooling layers are used to down-sample feature maps and reduce spatial dimensions. A GAP layer is applied after the final max-pooling layer. This approach allows the model to efficiently learn and retain essential information while reducing dimensions. The ReLU activation function is defined as ($F(x) = \max(0, x)$), which maintains higher values and sets negative ones to zero, enabling complex learning. The code snippet specifies the activation='ReLU' parameter while adding convolutional layers using Tensor Flow's Keras API. Incorporating max-pooling layers helps down-sample feature maps, reducing spatial dimensions while retaining crucial data. Max pooling selects the most significant

value within small regions of the input feature maps, effectively reducing dimensions while preserving relevant information. A GAP layer is introduced to Capture global features after the final max-pooling layer. The Global Average Pooling (GAP) layer serves a fundamental purpose: condensing feature map spatial dimensions while capturing global features. Following the GAP layer, the procedure entails conducting global average pooling across feature maps, yielding a single value for each channel. This operation reduces spatial complexity and encapsulates the overarching presence of acquired features in the input image. Consequently, the GAP layer generates a 1D vector, representing the globally averaged features extracted from feature maps. This vector subsequently serves as input for succeeding fully connected layers, where the extracted features are processed and applied to classification. Thus, the GAP layer's pivotal role lies in bridging convolutional layers and fully connected layers, furnishing a concise and informative depiction of input data.

In the proposed approach, Convolutional Neural Networks (CNN) is employed for feature extraction and classification, while the Extra Trees classifier is adopted as the classification algorithm. The architecture comprises initial and subsequent fully-connected dense layers, followed by the Extra Trees layer, which serves as the output layer for prediction. Additionally, a Competition layer is integrated, featuring four neurons Representing distinct classes. Various epoch counts for the Extra Trees classifier were tested, including 50, 100, and 200. However, after experimentation, it was found that setting the epoch count to 300 yielded higher training and validation accuracy. The learning rate was set at 0.001, and the input vector was initialized using random values. The decision to utilize 300 epochs considerably enhanced the model's accuracy during both training and testing stages.

6 Experimental results

In the experimental results, the proposed approach integrates Convolutional Neural Networks (CNN) for both feature extraction and classification, along with the Extra Trees classifier. The CNN model is initially trained to extract meaningful features from input images through Convolutional and pooling layers. However, it's important to note that the Extra Trees involve training by assigning epochs as the CNN model does. To address this, CNN is employed not only for feature extraction but also for classification, while the Extra Trees classifier is harnessed as the predictive model. This configuration allows for the generation of training and validation graphs, which are crucial for visualizing the learning process.

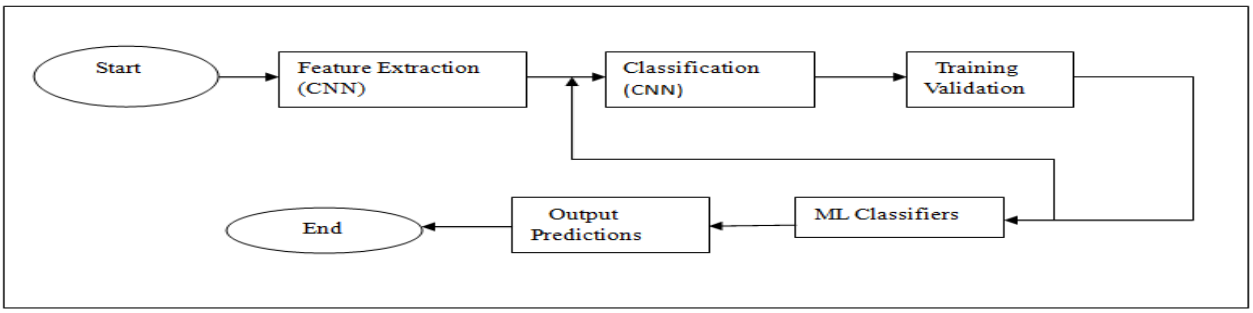


Figure 4: Flowchart of proposed method

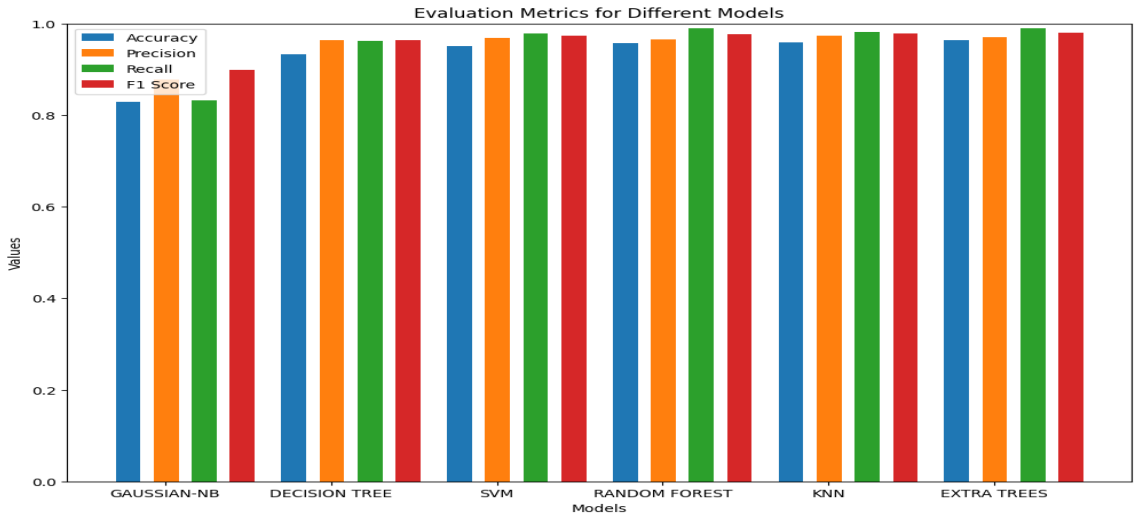


Figure 5: Performance of various ML classifiers

Once trained, the Extra Trees classifier predicts labels for the test dataset. The efficacy of the Extra Trees classifier is evaluated using assessment metrics like the confusion matrix and classification report. This combined methodology capitalizes on CNN's dual roles for feature extraction and classification, while the Extra Trees classifier contributes as the classification model. The complete methodology is given in Fig. 4.The various machine learning classifiers used in this model are SVM, Random Forest, Decision Tree, Gaussian-NB, KNN and Extra Trees. Out of all these mechanisms Extra Trees exhibits higher accuracy value. This setup synergizes the feature extraction and classification capabilities of the CNN model with the Extra Trees classifier, potentially leading to enhanced classification accuracy and overall performance on the test set.

From Fig. 4 it's evident that one approach involves utilizing the extracted features directly after the feature extraction step as input for the machine learning classifiers. Another method incorporates an intermediate step of showcasing training and validation graphs. In this method, the classification is applied subsequent to the feature extraction. The outputs obtained from the classification process serve as input for the classifier, leading to the final classification.

6.1 Assessment metrics

Evaluation metrics are essential tools for gauging the effectiveness of the proposed method, encompassing accuracy, precision, recall, and other factors. These metrics provide valuable insights for comprehensive assessment

Accuracy: It quantifies a classification model's correctness by comparing correct predictions to the total predictions

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Precision: It signifies true positives (TP) relative to the total of positive predictions (TP + FP):

$$\text{Precision} = TP / (TP + FP) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

Recall: It reflects the proportion of accurately classified positive samples among all actual positive samples:

$$\text{Recall} = TP / (TP + FN) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

F1-Score: It balances precision and recall for an accurate model evaluation:

$$\text{F1Score} = 2 * (\text{Precision} * \text{Recall}) / ((\text{Precision} + \text{Recall})) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

TP: In the context of classification models, TP represents the number of correctly predicted positive instances.

TN: It represents the number of correctly predicted negative instances.

FP: It represents the number of incorrectly predicted positive instances.

FN: It represents the number of incorrectly predicted negative instances. The Kappa coefficient (Cohen's Kappa) gauges agreement between raters or classifiers assigning categorical labels. It ranges from negative to positive, indicating agreement levels like slight, fair, moderate, Substantial, or almost perfect.

Mean Average Precision (MAP) is computed by averaging the AP values across all queries. SLIC is an algorithm used for super pixel segmentation, which involves dividing an image into compact, perceptually meaningful regions or segments. Accuracy is computed from SLIC.

7 Results and discussion

Various Machine leaning algorithms such as SVM, Decision Tree, KNN, Random Forest, Gaussian-NB, and Extra Trees are implemented for classification. Fig. 5 depicts the performance for various algorithms in terms of accuracy, precision, f1 score and recall. Extra Trees shows higher values when compared with other algorithms. Table.1. shows the values of the various metrics.

Table1: Summary of various machine learning classifiers

Model	Overall accuracy	Kappa	Precision	Recall	F1score
Extra Trees	0.9635	0.9621	0.9705	0.9904	0.9804
SVM	0.9508	0.9482	0.9685	0.9784	0.9739
RF	0.9579	0.9561	0.9655	0.9897	0.9775
KNN	0.9589	0.9572	0.9740	0.9816	0.9778
DT	0.9332	0.9284	0.9646	0.9628	0.9637
G-NB	0.8288	0.7935	0.8780	0.8330	0.8997

Table 1 provides a performance comparison of different models on the classification task. The evaluation metrics include overall accuracy, which measures the proportion of correctly classified instances, And the Kappa coefficient, serves as a metric to gauge the level of agreement or reliability between two raters when categorizing items into distinct groups. Kappa between 0 to 1 indicate agreement that is better than chance, with superior values indicating stronger agreement.

Additionally, precision, recall, and F1 score are provided to evaluate the model's performance concerning class-specific metrics. Higher values of these metrics generally indicate better model performance. From Table 1 it is inferred that Extra Trees exhibits higher accuracy of 0.9635 when compared with all other machine learning algorithms. The overall accuracy measures the proportion of correctly classified instances, and a higher value indicates better performance in terms of the total number of correct predictions. According to [13] the obtained kappa coefficient of Extra Trees is 0.9621 which is stated that lies within the range. The results

obtained from the trained models indicate the performance of the CNN-based feature extraction model and the Dense Neural Network classifier. The test loss and accuracy provide an overall assessment of the model's predictive capability on unseen data.

The confusion matrix reveals the model's performance for each class, identifying True Positives, True Negatives, False Positives, and False Negatives. The classification report presents metrics such as precision, recall, and F1-score giving insights into the model's performance across different classes. By analyzing these results, it is possible to evaluate the model's effectiveness, predict its strengths, and identify areas for improvement. The discussion of the results provides valuable insights into the model's performance, which can inform decision-making and potential applications in various domains.

The waveforms of CNN Extra Trees training and validation loss and accuracy are illustrated in Fig.6 and confusion matrix is shown in Fig.7.

7.1 Comparison with other studies

Comparing the outcomes of the proposed model with existing cases from the literature involves subjectivity. To facilitate this comparison, 13 recent studies are Extra Tree Techniques, as outlined in Table 2. Notably, the highest accuracy of 96.35% was achieved through the utilization of the CNN model coupled with the Extra Tree classifier.

7.2 Limitations and complexity of proposed model

The proposed method of utilizing a CNN for feature extraction and classification, followed by an Extra Trees classifier, offers a powerful fusion of deep learning and traditional machine learning techniques. However, this approach entails complexities and limitations. The intricate interplay between the CNN's learned features and the Extra Trees algorithm requires careful alignment and validation.

While the CNN's capacity to capture intricate patterns can be advantageous, potential challenges include hyper parameter tuning for both models, data availability for effective CNN training, and difficulties in interpreting CNN-derived features. Balancing these complexities and limitations is essential to harness the combined strengths of the two methods effectively.

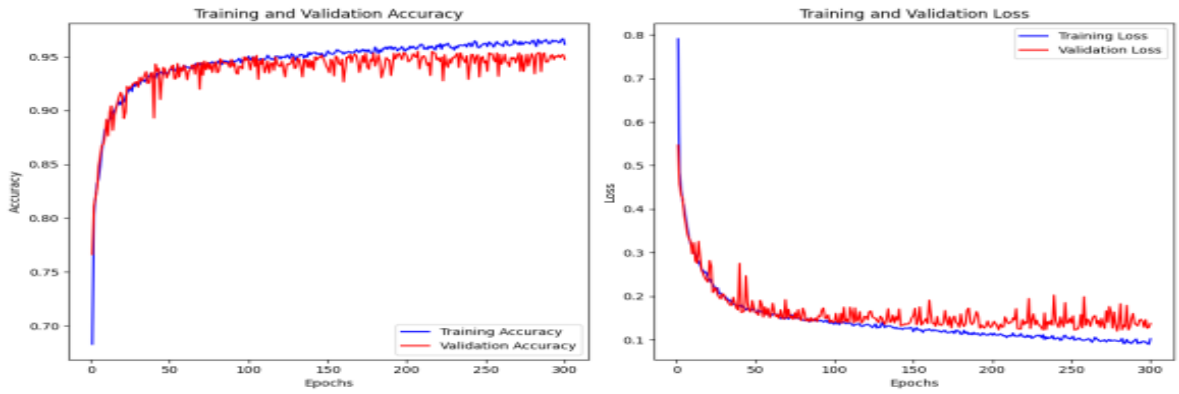


Figure 6: Wave forms of training, validation loss and accuracy of CNN Extra Tree

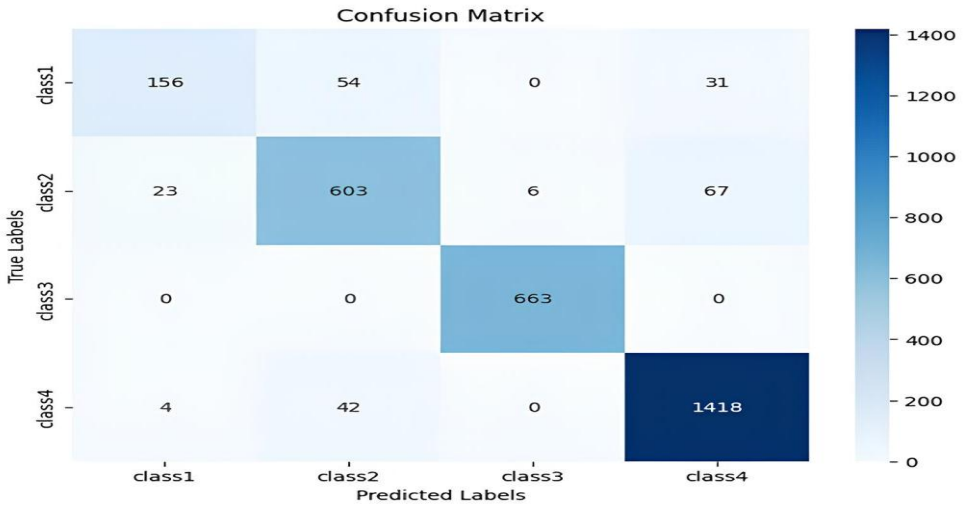


Figure 7: Confusion Matrix of CNN

Table 2: Comparison of proposed model with other studies from the literature

Model	Accuracy	Recall	F1- Score	Precision	Reference
Random Forest	82	93.3	92.1	-	[2]
Decision Trees	68	-	-	-	[4]
SVM	94	91	89	91	[6]
KNN	63	62	89	62	[6]
VGG16	92	92.1	52	92	[8]
SLIC-RF	91	99	91	100	[10]
Object-based	89	90	-	93.40	[11]
One-class	90	-	-	-	[13]
Single Shot Detector	84	80	78.5	81	[16]
SVM	66	-	-	-	[7]
RCNN	95	94.7	-	95.3	[18]
YOLO-V3	91	66	68	65	[17]
Relief-F	80	87.26	91.24	91.73	[19]
Extra Trees	96	99	98	97	Proposed method

8 Conclusion

The classifiers like Random Forest, Support Vector Machine, K-Nearest Neighbours, and Extra Trees are used in this paper. The proposed method of combining CNN for feature extraction, classification and Extra Trees as the classifier offers a promising approach for image classification and obtained an accuracy percentage of 0.9635. This method presents a versatile approach to image classification that merges the capabilities of deep learning and traditional machine learning. This strategy offers a promising solution for complex tasks, yet navigates challenges such as hyper parameter tuning, data availability, and interpretability. While demanding in terms of complexity, the model's potential to capture intricate patterns through the CNN's feature extraction and refine those using Extra Trees demonstrates its potential utility in tackling diverse classification problems. The Extra Tree classifier obtained an impressive accuracy of 96.35% along with an outstanding kappa coefficient of 96.21%.

Acknowledgement

I would like to extend deep sense of gratitude to the authorities of Acharya Nagarjuna University for the encouragement and support to do this work.

References

[1] Wu, Z., Chen, Y., Zhao, B., Kang, X., & Ding, Y. (2021). Review of weed detection methods based on computer vision. Sensors, 21(11), 3647 https://doi.org/10.3390/s21113647

[3] Yano, I. H., Alves, J. R., Santiago, W. E., & Mederos, B. J. (2016). Identification of weeds in sugarcane fields through images taken by UAV and Random Forest classifier. IFAC-PapersOnLine, 49(16),415-420 https://doi.org/10.1016/j.ifacol.2016.10.076

[4] Gao, J., French, A. P., Pound, M. P., He, Y., Pridmore, T. P., & Pieters, J. G. (2020). Deep convolutional neural networks for image-based Convolvulus sepium detection in sugar beet fields. Plant methods, 16, 1-12. https://doi.org/10.1186/s13007-020-00570-z

[5] Deng, Wei, et al. (2014): 26."Discrimination of crop and weeds on visible and visible/near-infrared spectrums using support vector machine, artificial neural network, and decision tree." Sensors & Transducers 26 https://www.sensorsportal.com/HTML/DIGEST/march_2014/Special_issue/P_SI_536.pdf

[6] Haq, MohdAnul. (2022). "CNN Based Automated Weed Detection System Using UAV Imagery." Computer Systems Science & Engineering42.2 https://doi.org/10.32604/csse.2022.023016

[7] Islam, N., Rashid, M. M., Wibowo, S., Xu, C. Y., Morshed, A., Wasimi, S. A., ... & Rahman, S. M. (2021). Early weed detection using image processing and machine learning techniques in an Australian chilli farm. Agriculture, 11(5), 387. https://doi.org/10.3390/agriculture11050387

[8] Jin, X., Che, J., & Chen, Y. (2021). Weed identification using deep learning and image processing in vegetable plantation. IEEE access, 9, 10940-10950.in a vegetable plantation." IEEE Access 9

[9] https://ieeexplore.ieee.org/abstract/document/9317797

[10] Tao, T., & Wei, X. (2022). A hybrid CNN–SVM classifier for weed recognition in winter rape field. Plant Methods, 18(1), 29. https://doi.org/10.1186/s13007-022-00869-z

[11] Brahim, J., Loubna, R., & Noureddine, F. (2021). RNN-and CNN-based weed detection for crop improvement: An overview. Foods and Raw materials, 9(2), 387-396. http://doi.org/10.21603/2308-4057-2021-2-387-396

[12] Kawamura, K., Asai, H., Yasuda, T., Soisouvanh, P., & Phongchanmixay, S. (2021). Discriminating crops/weeds in an upland rice field from UAV images with the SLIC-RF algorithm. Plant Production Science, 24(2), 198-215. https://doi.org/10.1080/1343943X.2020.1829490

[14] Gašparović, M., Zrinjski, M., Barković, Đ., & Radočaj, D. (2020). An automatic method for weed mapping in oat fields based on UAV imagery. Computers and electronics in agriculture, 173,105385. https://doi.org/10.1016/j.compag.2020.105385

[15] Razfar, N., True, J., Bassiouny, R., Venkatesh, V., & Kashef, R. (2022). Weed detection in soybean crops using custom lightweight deep learning models. Journal of Agriculture and Food Research, 8, 100308. https://doi.org/10.1016/j.jafr.2022.100308

[16] Bah, M. D., Hafiane, A., Canals, R., & Emile, B. (2019, November). Deep features and One-class classification with unsupervised data for weed detection in UAV images. In 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA) (pp. 1-5). IEEE. https://doi.org/10.1109/IPTA.2019.8936091

[17] Mellor, A., Haywood, A., Stone, C., & Jones, S. (2013). The performance of random forests in an operational setting for large area sclerophyll forest classification. Remote Sensing, 5(6), 2838-2856. https://doi.org/10.3390/rs5062838

[18] Arif, S., Kumar, R., Abbasi, S., Mohammadani, K., & Dev, K. (2021). Weeds detection and classification using convolutional long-short-term memory. https://doi.org/10.21203/rs.3.rs-219227/v1

[19] Yadav, P. K., Thomasson, J. A., Hardin, R., Searcy, S. W., Braga-Neto, U., Popescu, S. C., & Wang, T. (2023). Detecting volunteer cotton plants in a corn field with deep learning on UAV remote-sensing imagery. Computers and Electronics in Agriculture, 204, 107551. https://doi.org/10.1016/j.compag.2022.107551

[20] Veeranampalayam Sivakumar, A. N., Li, J., Scott, S., Psota, E., J. Jhala, A., Luck, J. D., & Shi, Y. (2020). Comparison of object detection and patch-based classification deep learning models on mid- to late-season weed detection in UAV imagery. Remote Sensing, 12(13), 2136., https://doi.org/10.3390/rs12132136

[21] Khan, S., Tufail, M., Khan, M. T., Khan, Z. A., & Anwar, S. (2021). Deep learning-based identification system of weeds and crops in strawberry and pea fields for a precision agriculturesprayer. Precision Agriculture, 22(6), 1711-1727. https://doi.org/10.1007/s11119-021-09808-9

[22] Mu, Y., Feng, R., Ni, R., Li, J., Luo, T., Liu, T., & Hu, T. (2022). A faster R-CNN-based model for the identification of weed seedling. Agronomy, 12(11), 2867 https://doi.org/10.3390/agronomy12112867

[23] Kiala, Z., Mutanga, O., Odindi, J., & Peerbhay, K. (2019). Feature selection on sentinel-2 multispectral imagery for mapping a landscape infested by parthenium weed. Remote Sensing, 11(16), 1892. https://doi.org/10.3390/rs11161892

[24] Dos Santos Ferreira, A., Freitas, D. M., da Silva, G. G., Pistori, H., & Folhes, M. T. (2017). Weed detection in soybean crops using ConvNets. Computers and Electronics in Agriculture, 143, 314-324. https://doi.org/10.1016/j.compag.2017.10.027

[25] Mishra, S., Mishra, D., Mallick, P. K., Santra, G., & Kumar, S. (2021). A Classifier Ensemble Approach for Prediction of Rice Yield Based on Climatic Variability for Coastal Odisha Region of India. Informatica, 45(3). https://doi.org/10.31449/inf.v45i3.3453

https://doi.org/10.31449/inf.v48i4.5845 Informatica **48** (2024) 663–684 **663**

Towards an Ontological-based CIM Modeling Framework for IoT Applications

Mohamed Bettaz[1], Mourad Maouche[2]
[1]Czech Technical University in Prague, Prague, Czech Republic
[2]Freelance Researcher, Constantine, Algeria
E-mail: bettamoh@cvut.cz, maouche805@gmail.com

Keywords: CIM modeling, requirement engineering, UFO ontology, IoT applications

Received: March 5, 2024

Few works addressed contributions of ontologies to Computation Independent Modeling (CIM) of Internet of Things' (IoT) applications. This work targets CIM artefacts developed using a combination of a goal-oriented requirements (KAOS) and a service-oriented (SoAML) modeling frameworks. This paper proposes an ontological-based framework intended to help CIM modelers in their preliminary analysis of IoT applications. We adopt the ontology reuse approach, an approach often used by the ontology engineering community, where specific ontology fragments are selected, adapted and/or refined, and merged. We use OntoUML to describe our fragments. The OpenPonk tool is used to edit and verify the syntax and the semantics of these fragments' models. The results of our contribution are summarized as follows. Improving the semantics carried by the metamodels of KAOS and SoaML modeling languages, through our proposed conceptualization grounded by the Unified Foundational Ontology (UFO), a sound ontological framework; setting a link between our proposed KAOS and SoaML ontology fragments; designing a (partial) IoT domain ontology to be integrated into our proposed CIM. An illustrative example, showing how to instantiate selected ontology fragments, demonstrates the applicability of our results to IoT applications.

Povzetek: Prispevek predlaga ontološko zasnovan okvir za modeliranje CIM (Computational Independent Modeling) IoT aplikacij. Uporablja pristop ponovne uporabe ontologij za izboljšanje semantike metamodelov KAOS in SoaML.

1 Introduction

Nowadays requirements' engineering, a discipline covering the analysis and the specification of systems under development, strongly relies on the Model Driven Engineering (MDE) methodology [1]. System analysis is concerned with the understanding and description of domains under examination and their business aspects (problem domain), whilst system specification is essentially concerned with the statement and the analysis of the prescribed requirements for the system under development (solution domain). Specific models, intended to capture domains (domain modeling view), business issues (business modeling view), and system requirements (requirement modeling view), put together constitute the so-called CIM [2]. A lot of research works conducted by the software engineering community have addressed and covered several aspects related to this kind of models. Some of them have been devoted to the analysis of IoT applications, services, and systems: for instance the authors of [38] proposed a dedicated UML profile intended to put together relevant computing and business modeling concepts, and the authors of [15, 9] suggested either an extension, or a combination of existing general purpose modeling languages.

Usually, modeling languages are supported by metamodels (models of models) intended to describe their abstract syntax and (structural) semantics. Models are supposed to be conform to the metamodel associated with the modeling language used to build them. CIM models described in appropriate modeling languages are intended to provide necessary descriptive (domain, business, and requirements) knowledge [2] that may be transformed into systems' implementation through a series of well-defined steps. Very often, these languages are thought and designed in terms of specific (combinations of) paradigms (goal-oriented, agent-oriented, task-oriented, and service-oriented paradigms for instance). Some of these paradigms may also share common kinds of modeling constructs. Moreover, the ontology engineering community also has conducted several research works investigating the potential contributions of ontologies to the Requirement Engineering [12] in general, and to the CIM modeling in particular [2]. Roughly speaking, "ontologies are formal explicit representations of shared conceptualization of parts of reality". They provide powerful descriptive means making them able to formally capture the essence of real-world phenomena relevant to various domains. We may mention in the following some of their potential links to MDE and also uses in the context of CIM modeling: Firstly, the theoretical work in [2] contributed to clarify the differences as well as the

inter-play between metamodeling, and ontological-based approaches to modeling languages. It states that, while ontologies are intended to capture only descriptive aspects of relevant parts of reality, CIM models are intended to capture both descriptive (domain, business) and prescriptive (requirements) aspects of systems under development. Secondly, the availability of worked-out ontologies covering not only the IoT domain itself (sensors, actuators, measurements, etc.), but also specific application domains where IoT technology may potentially be applied (health, transportation, business, and others), allows us to (re)use them (i.e., such ontologies) in IoT applications' CIM modeling (domain and business views). Thirdly, the literature reports various works showing the usage of ontologies in the context of MDE modeling languages. The authors in [20] raised the benefits of the use of ontological models as theoretical tools for analyzing and improving conceptual modeling languages, stating that "providing real-world semantics for modeling constructs" of conceptual modeling languages constitutes one of these expected benefits. Such conceptualizations of modeling languages may help to detect and make explicit hidden concepts carried by modeling languages, to derive solid metamodels for new developed modeling languages, or to support model transformations' activities thanks to a set of well-known ontology engineering tasks (integration, mapping, and merging of ontologies). Basic knowledge on these tasks is given in Section 3. The wide and ever-growing deployment of both ontologies, covering aspects of the real life and the computing discipline on one side, and IoT applications and services on the other side, motivates our interest into this subject. To the best of our knowledge, few works have been specifically devoted to the use of ontologies to capture the semantics of CIM models associated with IoT applications under development. The reported works mainly addressed the use of ontologies for Ambient Assisted Living (AAL) applications' development [13, 11], a particular application domain where IoT technology is applied. The objective of this paper is to propose an ontological-based framework intended to help CIM modelers in their preliminary analysis of IoT applications under development. This is achieved through the construction of a global ontological model regrouping and relating IoT domain key concepts, application domain concepts, and concepts associated with key constructs offered by the used CIM modeling languages. Instances of this global ontological model can be analyzed, simulated, and validated using appropriate tools. Ontologies' merging is the employed mechanism to build the intended global ontological model [35]. More concretely we propose to:

(a) Employ KAOS (Knowledge Acquisition in Automated Specification of Software Systems) for the goal-oriented aspects [46], and SoaML (Service-oriented architecture Modeling Language) for the business view [14]. These choices are mainly motivated by the availability of many works reporting on ontological views of, respectively the goal-oriented and service-oriented paradigms.

(b) Build preliminary ontological models for KAOS and SoaML respectively, thus providing real-world semantics for their respective key modeling constructs. The ontological models associated with these modeling languages are built by exploiting the knowledge carried explicitly by their metamodels, and also the informal knowledge carried in their textual documentations [24].

(c) Employ ontological models to represent and describe CIM domain views associated with IoT applications. This is mainly motivated by the availability of several ontological models representing the IoT domain (sensors, actuators, etc.), and a lot of ontological models representing various application domains.

(d) Construct an ontological model intended to represent a global conceptual view regrouping and relating in a "pragmatic" way the three CIM views of IoT applications. This is achieved by merging the ontological models mentioned in (b) and (c). The merging operation is done in such a way that the ontological models to be merged, seen as constituent fragments, are kept unchanged in the resulting merged ontological model.

It is worth mentioning that our ontological models rely on the UFO infrastructure, and are expressed in OntoUML, a UFO ontology-based modeling language. Basic knowledge on OntoUML and its supporting UFO foundation is included in Section 3, while the motivation behind the decision of such choices is exposed in Section 4.

The rest of the paper is organized as follows. Related works are presented in Section 2. A preliminary knowledge on ontologies, OntoUML, UFO infrastructure, Goal Oriented Requirement Ontology, and a set of relevant modeling languages is summarized in Section 3. The method used to carry out this work as well as the motivations behind the choice decision is explained in Section 4. Section 5 exposes and details our proposed UFO-based ontology fragments for the KAOS framework. Section 6 presents our proposed reinforcement of the business dimension for the KAOS framework. A preliminary SoaML ontology is described in Section 7. Section 8 presents and explains the approach used to merge our KAOS and SoaML ontologies. In Section 9, we show how to transform OntoUML models into appropriate gentle UFO (gUFO) classes aimed at their computerization. Section 10 presents an illustrative example demonstrating the applicability of our results to IoT applications. Concluding remarks, limitations, and future works are outlined in Section 11.

2 Related works

The use of ontologies in the software engineering discipline has been investigated for over a decade addressing its different fields with various motivations and intentions. Ontologies for CIM modeling of systems in general, and of IoT-based systems specifically have also been subject of

research interests and works. Table 1 encloses the results of related works. It summarizes key concepts used in these works and shows that our proposal is not addressed in them.

2.1 Ontologies for CIM modeling

The main contribution of [13] consists in a "smooth" combination of various existing ontology fragments covering the goal concept using a variant of an i-star (also denoted i*) [16] based ontology, and other AAL related domains such as IoT (sensors, measures, etc.) and domotic. Their objective is to propose a goal-oriented ontology-based methodology for AAL systems motivated by the need of building AAL knowledge sources allowing the access to stored or inferred relevant knowledge. Our contribution, whilst similar to the work in [13], differs in three aspects; (i) it includes a new modeling dimension, namely the business view, hence covering all aspects of CIM modeling; (ii) we adopt KAOS rather than i-star. Both (goal-oriented requirement engineering frameworks) present benefits and drawbacks [45]. The KAOS ability to distinguish between agent roles is one of the motivations behind our choice of KAOS; (iii) our proposed ontology is built on UFO, a solid foundational top-level ontological framework. The work in [11] consists in building an ontology, called Goal Service Ontology (GSO), as a support for designing a framework for dynamic service discovery, composition, and use. Although GSO is not dedicated to IoT applications, it has been applied to the modeling of AAL applications. GSO, as our ontology, is built on refining UFO-C ontology fragments for the goal and tasks (an operation-like concept). Contrarily to our contribution whose objective is to build an ontology to formalize CIM models using a combination of KAOS and SoaML, GSO is intended to back the metamodel of a new designed DSL language. Lastly, our conceptualization of the service paradigm relies mainly on refinements of pieces of UFO-A [17] and UFO-B [21] ontology fragments rather than on the UFO-S [30] ontology fragment as in GSO. The work in [43] consists in adopting an ontology approach for the modeling of SysML/KAOS domains. Roughly speaking, a metamodel for domain modeling is built on some OWL (the web ontology language) model elements, together with new added model elements. In our work, domains are modeled as pieces of refined UFO-A categories smoothly combined with our proposed KAOS and SoaML ontologies. Go4SoA [10] is a proposal combining the goal with the Service-oriented Architecture (SoA) paradigm. The approach consists in extending and enriching the SoaML metamodel with the goal concept. Regarding our work, the combination is done in an ontological way, by bridging KAOS and SoaML ontologies. [3] treats on the "semantic interoperability across IoT domains in cross-domain applications".

2.2 Ontologies for KAOS and SoaML

The literature reported ontology proposal(s) for KAOS [27], and for SoaML [30, 37, 29]. In [27], the authors used on the one hand a standard KAOS metamodel as a reference to build their KAOS ontology, and on the other hand the Unified Enterprise Modeling Language (UEML) as an ontological modeling language. UEML is usually intended for enterprises and information systems' modeling. Our KAOS ontology, not only refers to a standard KAOS metamodel, but also it is based on proposed refinements of modeling elements drawn from the UFO-C (Agent and Goal) ontology fragments [19, 23], and also on proposed refinements of the Goal-oriented Requirement Ontology (GORO) [31, 23]. Regarding SoaML, [30] and [37, 29] are among the recent works aiming at introducing ontology concepts for SoA and SoaML. The authors of [30] have analyzed and evaluated the use of UFO-S, an UFO sub-ontology for services, in various approaches including the SoaML one; those of [37] have proposed a set of ontologies covering the general aspects of the so-called Service Engineering (service-oriented architecture, software service ontology, etc.) including ontologies for SoaML. Their work relies on the Open Group Service Ontology, and also on the ISO/IEC SoA Reference Architecture. [29] provides a comparison between the UFO-S service ontology and other similar service ontologies including the Open Group Service Ontology.

Our SoaML ontology is built using refinements and adaptations of some modeling elements of UFO-A and UFO-B ontology fragments. The suggested adaptations mainly rely on relevant knowledge drawn from the OMG SoaML specification document [33] as well as from [37, 29].

3 Preliminary knowledge

In order to make our paper (as much as possible) self-contained, this section will summarize basic knowledge on ontologies, GORO, and various relevant modeling languages (KAOS, OWL and RDF/RDFS, and SoaML).

3.1 Ontologies

Ontologies have been widely used in various disciplines and real-world domains. The authors in [26] state that "in computer science, ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. Ontologies are used to conceptualize domains and to reason about domain properties". Ontologies are often characterized by their level of generality. General classes are specified in top-level ontologies, whilst more specific classes are described in lower-level ontologies [18]. Top-level ontologies are intended to describe very general concepts. Some of these ontologies are qualified as reference ontologies in the sense that they are dedicated to a specific domain and recognized as a reference by its relevant communities. UFO

Table 1: Summary table

Ref.	Proposal	CIM Requirement	CIM Business	CIM Domain	Ontology / Metamodel	Application	Remark
[38]	IotReq	KAOS-like goal	SoaML services	SoaML and UML	UML profile	IoT app.	no use of ontology
[11]	GSO	goal, agent, task	services (client, provider)	UML class models	UFO grounded	includes IoT app.	ontology for new DSL
[10]	Go4SoA	goals	BPMN	UML class models	extended metamodel with goal	bus. SoA oriented app.	no use of ontology
[44]	SysML/ KAOS domain model.	KAOS/ SysML method	…	ontol. based domain models	domain knowl. as an ontology	CPS; IoT-based systems	not fully integr. in an ontology
[13]	An ontol. based methodol. for AAL	Requirem. as GRL goals	…	mix. frag. from various dom.	GoAAL an ontology for AAL app.	AAL systems	specific
[27] [32] [45]	KAOS model. methodol.	goals for requirem. model.	…	domain model. with UML	metamodels & ontology	general purpose; mainly CPS app.	missing bus. aspect model.
[29] [30] [33] [37]	SoaML model. methodol.	from bus. proc. modeling.	…	UML class diag. & profile	UML profile	bus. oriented app.	missing goal approach for requirements

is a formalized top-level reference ontology [17]. It has been successfully used to conceptualize a lot of specific domains, and particularly the domain of modeling languages [20, 22]. UFO includes four ontology modules: UFO-A, intended for the conceptualization of Endurants (commonly called Entities) [17], UFO-B for the Perdurants (commonly known as Events) [21], UFO-C for the Social and Intentional Objects [19], and UFO-S for Services [30]. OntoUML is an UFO-based UML profile, formally conceptualized by the UFO reference ontology [40]. It provides a set of class stereotypes such as Category, Kind, Subkind, Relator, Events and their related "association relationships" stereotypes (cf. [41] for more details on OntoUML specifications). OntoUML, mainly used in the ontology engineering field, has retained the attention of the software engineering community [36]. In particular, [6, 5] refer to a newly OntoUML-inspired language that is intended to augment the object-oriented formal language Object-Z with OntoUML-like features. Ontology integration, merging, and mapping are among the interesting topics discussed by the ontology engineering communities; these topics have been investigated for more than two decades. Recent research surveys [35, 39] discussed the confusion about these ontology operations and their misuse.

Furthermore, ontologies may be seen as implementations of knowledge based systems. Thus they may be populated (instantiated) and queried as databases.

3.2 RDF and RDFS

OWL is an extension of Resource Description Framework (RDF) for building ontologies. RDF is a framework for describing resources organized as data graph models. An RDF statement (rdf:<statement>) is a triple (subject, predicate, object), where subjects and objects are graph nodes and predicates are graph edges. Internationalized Resource Identifiers (IRIs) are used to identify nodes and edges of an RDF graph according to user defined namespaces. An IRI described by the symbol ":" preceding a string means that a default namespace is used. Among RDF syntax notations we may mention those that define the elements of triples (rdf:subject, rdf:predicate, rdf:object) and RDF properties (for example rdf:type employed to indicate that a given named resource is an instance of a class). RDF Schemas (RDFSs) is a vocabulary extension of RDF that allows the construction of taxonomies of classes and properties. Among RDFS syntax notations we may mention those that define a class of classes (rdfs:Class), those that declare a subject as a subclass of a class (rdfs:subClassOf), and those that declare the domain and range of a subject property (rdfs:domain, rdfs:range). SPARQL, a "SQL-like" query language, is widely used for querying RDF graphs.

3.3 GORO in a nutshell

GORO is an ontological framework intended to conceptualize a well-established requirement engineering approach

called Goal Oriented Requirement Engineering (GORE) [31, 4]. Several notions and relationships related to the requirement engineering discipline are identified and semantically defined. However, GORO is used in this work only as a reference facilitating the understanding of the basic key GORE notions. One of the objectives of GORO is to enable the interoperability between different GORE modeling languages [4]. It is worth mentioning that the KAOS modeling approach supports the GORE approach.

3.4 The KAOS modeling framework

KAOS is a well-known requirements' engineering framework, supported by a set of modeling constructs, that primarily relies on the goal paradigm to specify system requirements. It has been used in the industry by software as well as system engineering communities either as a single modeling language [25, 42] or combined with other modeling languages [34]. More concretely, a KAOS specification consists of four inter-related models: a goal model, a responsibility model, an operation model, and an object model. The goal model is intended to describe the intentions of actors (stakeholders and software). It is organized as a hierarchy of goals representing a set of strategic (high-level and middle-level goals) and operational (leaf-level goals) intentions as well as their relationships. KAOS identifies two sorts of operational goals, named requirements and expectations. The distinction between these two sorts of operational goals is illustrated in Section 5. The responsibility model describes the way the elicited operational goals are assigned to specific actors, namely environment agents (stakeholders) and software agents, for achievement purpose. The operational model concerns the operationalization of the elicited leaf goals in terms of the operations performed by the assigned agents . The object model is intended to model elicited agents, and elicited application domain entities to be monitored (through sensors) and/or controlled (through actuators) [15]. More details on key KAOS modeling constructs are given in Section 5. In this work, we propose a preliminary UFO-based ontology intended to conceptualize a subset of the KAOS key modeling constructs, that are based on the KAOS metamodel reported in [32, 44]

3.5 The SoaML modeling language

SoaML is a modeling language supporting the Object Management Group (OMG) Services' Reference Model [33]. SoaML has been used in industry and enterprises by software engineering and business communities. It mainly focuses on the service paradigm where business and software services are treated in a uniform way. SoaML provides business experts with means allowing them to describe business issues, and it provides software engineers with means allowing them to model computing issues (software). Its key modeling constructs, provided by a specific UML profile, capture relevant SoA entities such as service, service contract, service interface, service architecture, agent, participant, capabilities, operations, and others. More details on these modeling entities are given in Section 7. A UML metamodel is associated with this profile. It is worthwhile to mention SoaML4IoT, an extension of the SoaML profile, intended for IoT applications [9]. It refines some of the (native) SoaML model elements and introduces new ones. In this work, we focus on SoaML and propose a preliminary UFO-based ontology intended to conceptualize a subset of the SoaML key modeling constructs, based on the SoaML metamodel reported in [33].

4 Method

The objective of this work is to propose a framework aiming at conceptualizing IoT CIM models in an ontological way. For this purpose, we followed two steps.

1. State the starting point
   Two alternatives are to be considered: CIM aspects are expressed either directly by using available appropriate modeling languages or in a modeling language-neutral. The advantage of the first alternative is that the literature reported a number of modeling languages that have already explored, identified and captured (through metamodels) a large set of key concepts and an almost "shared" vocabulary related to the whole CIM aspects. The issue is now to select, among the available languages, the ones that not only cover the different aspects of CIM models but also have been successfully for modeling IoT requirements.

2. Build the targeted ontology (fragments)
   The ontology engineering community suggests two alternative methods to build ontologies. Either constructing ontologies from scratch starting from well-defined requirements (for instance the so-called Competency Questions), or by reusing available ontology fragments. Ontology integration and ontology merging are the two well-known methods adopted in the ontology reuse approach. Ontology integration consists in reusing selected and evaluated ontology fragments, extending them, customizing and adapting them to the targeted context. Ontology merging consists in merging fragments that capture the same "reality" or share some common concepts. In our work we used both integration and merging methods in an ad-hoc way.

4.1 Selection of appropriate CIM modeling languages

Usually, CIM modelers select and combine specific languages to cover the three CIM aspects (i.e., domain, business, and requirements aspects). In this work, ontologies, due to their high descriptiveness power, are used in conjunction with suitable languages to improve the modeling

of CIMs. We retain KAOS for the modeling of the requirements as well as for the modeling of IoT and specific application domains, and SoaML for the modeling of the business aspect. This work proposes to build, and merge ontologies associated with each of these two languages. Two main reasons motivate the choice of KAOS. Firstly, it has shown its effectiveness in the modeling of the requirements of real-life and academic projects. It has been also used, in combination with the SysML language [15] for the development of (engineered) systems in general and AAL systems in particular; these kinds of systems are characterized by a strong use of sensors and actuators. Secondly, we can reuse and, by the way, build on available relevant ontology fragments ([19], [31], [4], [27], [23]). Three reasons motivate the choice of SoaML. Firstly, due to its support for SOA, SoaML provides means to model business issues as well as software issues, thus making it not only able to support CIM business views but also to offer a smooth bridge to software design views. Secondly, a previous work [11] and a more recent one [10] emphasized the use of the service paradigm in an ontological-based approach for respectively the AAL systems, and IoT systems. Thirdly, as for KAOS, the availability of ontologies intended to conceptualize the service paradigm ([30]] as well as SoaML ([37], [28]) allows us to work-out our preliminary ontology for SoaML without starting from scratch.

4.2 Building of our ontology fragments

For this purpose we followed two steps.

1. Construction of preliminary KAOS and SoaML ontologies
   Nowadays, new ontologies are mainly built by reusing (parts of) available ontologies. The ontology integration methods requires to select, evaluate and customize available ontology fragments matching with the reality to conceptualize. On one side, for the purpose of our KAOS ontology, we selected well-evaluated fragments provided either by UFO or GORO. It is worthwhile to note that the GORO ontology allowed us to get a better understanding of the concepts inherent to the goal-oriented requirement engineering. On another side, our proposed SoaML ontology fragments are mainly inspired from a set of Service and SoA ontology proposals reported in the literature ([29], [30], [37]). These works allowed a better understanding of key concepts of SoA and SoaML.
   Finally, the adaptation and customization of the selected fragments to KAOS and SoaML is mainly governed by an appropriate mapping of their concrete key constructs in conformance with their associated published metamodels. For convenience and uniformity purposes, the proposed ontologies are expressed using a unique ontology modeling language, namely OntoUML. This modeling language has been chosen because it is backed up by a theoretical reference ontology that puts solid foundations for the goal, agent, and service paradigms.
2. Merging our worked-out preliminary ontologies
   As a first attempt, we chose a pragmatic approach to conduct the merging of our proposed preliminary ontologies. A mapping operation stating a direct or indirect correspondence between similar constructs is required before performing the merging. We start by identifying potential similarities between shared modeling constructs, then we either merge both equivalent concepts into one concept in case of a direct mapping, or we build an intermediate ontology serving as a bridging ontology in case of an indirect mapping.

5 Ontology fragments for KAOS

5.1 Choice of UFO and OntoUML

To the best of our knowledge, the sole attempts to build an ontology-based model for KAOS is in [27], using the Unified Enterprise Modeling Language (UEML). This language, intended to unify enterprise modeling and information system modeling, is backed-up by the Bunge-Wand-Weber (BWW) model and Bunge's ontology. Among its use cases, we can mention its use to describe the metamodel of KAOS [27]. The authors of this work noticed that UEML is difficult to use, and exposed some limitations of the UEML language itsellf as well as its ontological framework (at the date of the publication of their work). In our work, we chose to use OntoUML (an ontological-based version of UML) mainly for the following reasons: OntoUML is backed up by UFO, a formal top-level foundational ontology presenting good relevant ontological properties such as completeness; availability of reliable middle-level UFO ontology fragments conceptualizing in a very abstract way concepts like goal, agent, event, and others; OntoUML is easy to learn and use because of its proximity with UML (widely used in various communities); OntoUML offers a palette of stereotyped classes and relationships able to cover and face numerous and various modeling situations. We used the OpenPonk tool [8] to edit and verify the syntax and the semantics of each of our models. The semantics is checked against well-defined anti-patterns.

5.2 Approach

The scope of this work encompasses the four KAOS models constituting the KAOS framework (Agent, Goal, Operation, and Object). The concepts of Obstacle/Conflicts relating to Goal modeling are not addressed, because not pertinent to the results of this (part of) work. Our proposal builds on and refines specific UFO ontological elements to conceptualize concepts relevant to the KAOS models, according to the KAOS universe of discourse. The KAOS metamodel is used as a primary source to capture this universe of discourse, augmenting by the way the degree of validity of our ontology fragment. GORO ontology

has also served as a reliable source for this work, however GORO is mainly used for comparing similar goal-oriented requirement frameworks (KAOS, i-star, and others). Thus, GORO is built in such a way that it abstracts features specific to each framework. Various UFO ontology fragments give a solid theoretical conceptualization of the goal and agent concepts, and their inter-play as well. Relevant UFO classes and relationships of these fragments are selected, reused, and then refined in such a way to be the most closest as possible to the KAOS constructs. The following exposes, for each model of the KAOS framework, the aspects that will be emphasized by our ontology fragments.

- **KAOS Agent Model**
  The KAOS agent fragment has to conceptualize the following aspects:
  (a) KAOS agent, its different sorts as well as the specificity and the meaning of each of its sorts (conformance with the metamodel).
  (b) The eventual real world entities that may be behind each sort of agent. These aspects are out-of scope of the KAOS metamodel. We added them because, in this work, we are interested in the IoT applications.
  (c) The relationships (and their nature) that capture the links between the entities mentioned in (b) and their corresponding agents elicited in (a).

  Figure 1 depicts the ontology fragment that captures the (a) and (c) aspects, while Figure 2 and Figure 3 depict, respectively, the ontology fragments that capture the aspect (b).

- **KAOS Goal Model**
  The KAOS Goal ontology fragments have to conceptualize the following aspects:
  (a) KAOS goal, and its different forms as well as the specificity and the meaning of each of its forms (according to the metamodel).
  (b) The way KAOS goals are classified, structured and organized.

  Figure 4 together with Figure 5 depict the ontology fragment that captures the aspects mentioned above .

- **KAOS Operation and Object Model**
  The KAOS Operation and Object ontology fragments have to conceptualize the following aspects:
  (a) The KAOS operation and their different sorts.
  (b) The operation parameters.
  (c) The pre-state and post-state attached to the operations.
  (d) The Object model state on which operations apply.

The fragment related to these aspects is not described in this paper. All the above mentioned figures are detailed in the following sub-sections.

5.3 KAOS agent ontology

The KAOS metamodel identifies and reveals in an explicit way two sorts of agents according to the role they play in the context of a KAOS based requirements' model, namely an environment agent and a requirement agent. KAOS environment agents are intended to enforce the "satisfaction" of the so-called expectations, whilst KAOS requirement agents are intended to enforce the satisfaction of the so-called "requirements". Expectations and requirements are specific KAOS goals explicitly defined in the KAOS metamodel. KAOS environment agents are exemplified by software agents (artificial active entities) that are concepts related to the computing context, and KAOS environment agents are exemplified by either humans /corporations (physical active entities) that are concepts related to the business and technical contexts. While the previous sentence is just an informal statement written in KAOS documents, our ontology makes it explicit. Figure 1 shows four sorts of OntoUML stereotyped classes (Category, Kind, Relator, and Role). The stereotype Category is generally used for representing abstract classes. We use them to glue our ontology to relevant UFO top- and middle-level categories and then specialize them into for instance Kind, or other stereotyped rigid sortal entities. KAOS agents are characterized by two features. On one hand they are agents (active entities), on the other hand they are intentional agents belonging to the KAOS framework. Figure 1 shows that the KAOS Agent concept is represented by a class (stereotype Kind) that specializes the abstract class Category Agent. This way we ensure that our ontology fragment (Figure 1) faithfully captures the intentional agent concept. Because these agents belong to the KAOS framework, we need to conceptualize the fact that the KAOS framework reveals two sorts of agents. Here we have the choice to capture this aspect by specializing the Kind Agent either into two subkind agents or into two role agents. However the nature of these two sorts of agents seems to be more close to Perdurants rather than Endurants (according to the UFO terminology). That is why we retain the stereotype Role rather than the stereotype Subkind to deal with this aspect. Moreover, Figure 1 captures also the informal statement related to the real-world entities that may exemplify these two roles. For this purpose, we introduce two relators. The first one, Requirement Agent Reification, conceptualizes the idea that software agents, through their specialized role (RoleMixin) KAOS IoT Software Agent, exemplify KAOS Requirement Agent (RoleMixin). The second relator, Environment Agent Reification, conceptualizes the idea that Stakeholder, through their specialized role (RoleMixin) IoT Application Stakeholder, exemplify KAOS Environment Agent.

5.4 Stakeholder ontology fragment

The following fragment, which is outside of the scope of the KAOS metamodel, is useful and relies on the following UFO knowledge: UFO ontology defines a category, named Substantial, which is specialized into the Object and

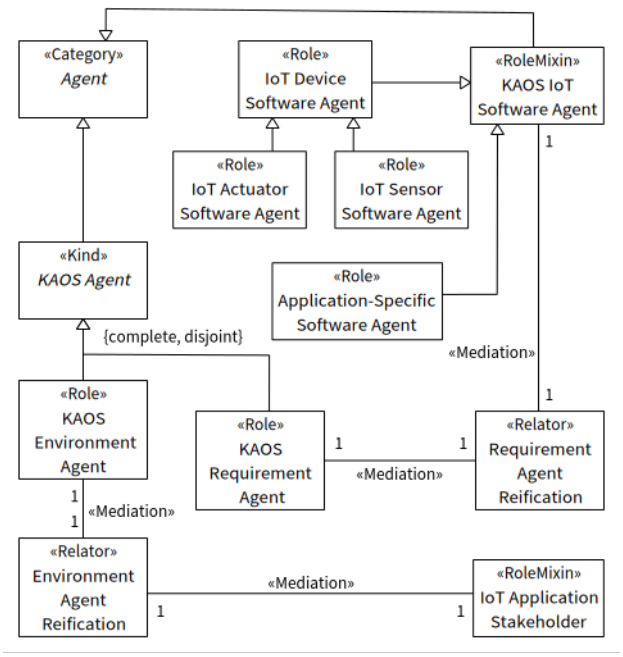


Figure 1: KAOS agent ontology

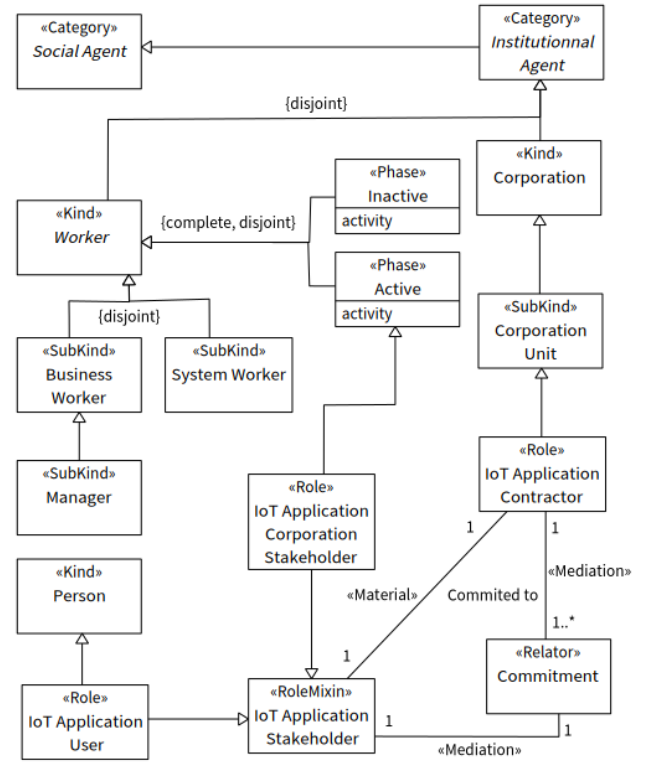


Figure 2: Stakeholder ontology

the Agent categories. Agents conceptualize active entities, whilst objects conceptualize passive and unintentional entities. Agents are of different sorts: physical (e.g., machine, human being), artificial (e.g., software), and social agents (e.g., society associations, institutions, stakeholders). Social Object, a refinement of UFO Object, is intended to conceptualize social objects like value, money, etc. In Figure 2, we propose a conceptualization of stakeholders in the context of IoT applications. Category Institutional Agent, a specialization of UFO Social Agent, is intended to abstract all sorts of institutional agents. Figure 2 shows two sorts of concrete institutional agents. The first one represents workers, conceptualized by a Kind class, acting as IoT project stakeholders, and the second one represents corporations, conceptualized by also by a Kind class, acting as initiator and/or developer of IoT applications. Corporations (Kind class) are modeled as a group of corporation units (SubKind class). Workers may be eventually engaged or not in IoT projects. Two disjoint Phase classes (Active, Inactive) allow to conceptualize this situation. Moreover we distinguish three sorts of workers: Business and System, modeled as Subkind of Kind Worker, and Manager modeled as Subkind of Business. The fragment includes also, a Role class, intended to conceptualize users of IoT applications (Role IoT Application User specializing Kind Person). IoT Application Stakeholder is intended to model the fact that the IoT application stakeholder notion includes application users as well as active workers. Because IoT Application Stakeholder is aggregating two Roles (customer, worker), RoleMixin is the more appropriate OntoUML stereotype class to represent it. Finally, the Relator Commitment conceptualizes the fact that a commitment links the IoT application stakeholders with the with the corporation units that are contractors with the project, i.e., the corporation unit that owns the project and also the corporation units such as the Ambulance Supplier Center, the Security Center, and the Home CareGiving Center. The IoT Application Corporation Stakeholder class represents the active corporation workers participating into the IoT application project.

5.5 IoT software agent ontology fragment

Figure 3 depicts a software agent ontology fragment. In the context of IoT applications, we identify two sorts of IoT software agents: those which are embedded into IoT devices, and those which are hosted by computing devices like servers and others. The first sort is modeled as a Subkind class (IoT Device Software Agent), and the second one as a Role class (Application-Specific Software Agent). The Subkind IoT Device Software Agent is specialized into two Role classes (IoT Sensor Software Agent and IoT Actuator Software Agent).Three Relators (S-Embedding, A-Embedding, and Hosting) conceptualize the embedding as well as the hosting of software agents according to their respective digital devices. Digital Device, is itself specialized into two subkinds: Subkind IoT Device and Subkind Computing Device.

5.6 KAOS goal model ontology

The proposed KAOS goal model ontology is composed of two ontology fragments. The first one, depicted in Fig-

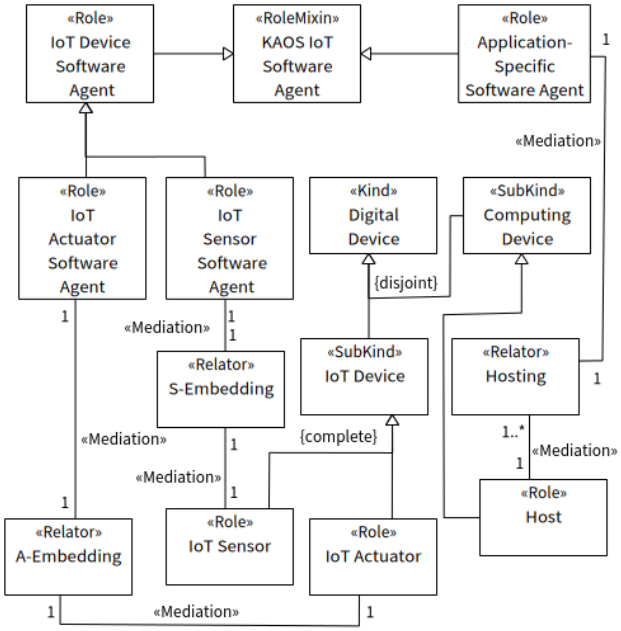


Figure 3: IoT Software agent ontology

ure 4, models the KAOS goal concept as well as the hierarchy structure of the KAOS goal model. The second one, depicted in Figure 5, conceptualizes the various sorts of KAOS goals as well as the relevant links to their corresponding KAOS agents.

5.6.1 Goal and hierarchical model

Figure 4 presents an ontology fragment including three parts: the first part represents the KAOS goal concept, the second part the hierarchical structure of the KAOS goal model, and the third part the concept of the KAOS abstract goal (decomposable goal).

- *Goal Concept*
  A KAOS goal is seen as a RoleMixin of the Category Requirement Engineeging Goal Specializing the Category Goal. Figure 4 shows, for instance, the RoleMixin i^* Goal representing the concept of goal in the i^* framework. The Category Requirement Engineering Goal is introduced to factorize abstract shared features that may exist between different frameworks that deal with the goal oriented requirement approach.

- *Goal Hierarchical Model*
  The RoleMixin Retained KAOS Goal in Figure 4, a specialization of the RoleMixin KAOS Goal, conceptualizes the fact that an elicited KAOS goal is retained to be inserted into the hierarchy of a goal model. The hierarchy concept is modeled as a Kind (Goal Hierarchy), a refinement of an abstract Category, named Structure, intended to represent all sorts of structures (not represented in the figure). The Relator Incorporation, a relationship between a retained KAOS (RoleMixin Retained KAOS Goal) and a KAOS goal hierarchy (RoleMixin KAOS Goal Hierarchy) that conceptualizes the fact that a retained goal is inserted into a goal hierarchy. According to its position in the goal hierarchy, a retained goal may play either the role of an abstract goal or the role of a concrete goal. Figure 4 includes a Kind (Tree Node) to represent nodes of the Tree structure. Tree Node is used as a secondary class allowing the modeling of abstract goals incorporated into a non-leaf node of the hierarchy, and the modeling of a concrete goal incorporated into a leaf node of the goal hierarchy.

- *Abstract Goal*
  Contrarily to a KAOS concrete goal, an abstract one may be refined into sub-goals (decomposable). The KAOS metamodel emphasizes two sorts of goal refinement: AND refinement and OR refinement. The AND refinement specifies that a refined goal is achieved if and only if all sub-goals are achieved. The OR refinement specifies that a refined goal is achieved if and only if one of its sub-goals is achieved. For this purpose, Figure 4 includes two roles (Role AND Decomposable and Role OR Decomposable) refining the RoleMixin Abstract Goal. Finally two relators (Relator OR Parent and Relator AND Parent) conceptualize the relationships between an abstract goal (as a parent goal) and its sub-goals (as child goals). To this end, Figure 4 includes two roles (Role AND Child and Role OR Child) which are refinements of the RoleMixin Retained KAOS Goal.

Figure 5 depicts an ontology fragment intended to represent three sorts of KAOS concrete goals defined in the KAOS metamodel, as well as their relationships with their corresponding KAOS agents or with the KAOS object model. KAOS concrete goal sorts are often named realisable or operational goals in the KAOS literature.

The RoleMixin (KAOS) Concrete Goal is specialized by three roles (Role Expectation, Role Requirement, and Role Domain Property) intended to represent the three sorts mentioned above. Expectations are taken in charge by KAOS Environment Agents, whilst Requirements are to be made operational (realizable) by KAOS Requirement Agents. For this purpose, Figure 5 includes two relators (Relator Assignment and Relator Responsibility) for the conceptualization of the (above mentioned) relationships between Role Expectation and Role KAOS Environment Agent on one hand, and between Role Requirement and Role KAOS Requirement Agent on the other hand. The Role Domain Property is specialized by the Role Domain Hypothesis and the Role Domain Invariant. Domain hypothesis represents hypothesis on the state of the object model that must hold. They are supposed to be enforced by the environment of the application domain. For this purpose the Relator Hypothesis Enforcement conceptualizes the relationship between the Role Implicit Enforcer and the Role Domain Hypothesis. The Kind Environment is modeled as a (shared) part of

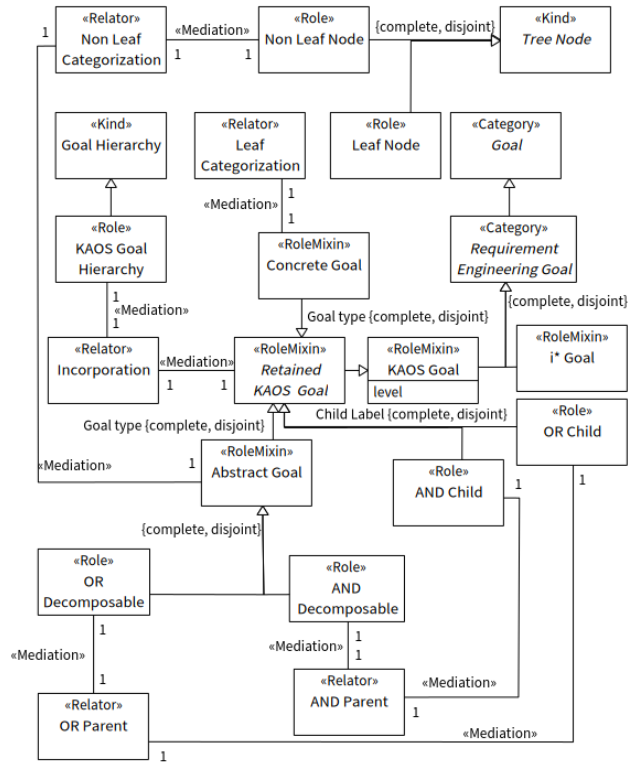


Figure 4: KAOS goal hierarchy ontology

the Kind Application Domain. The domain invariants represent invariants that must always hold in any state of the application domain. The application domain must always fulfil them. For this purpose, the Relator Fulfilment conceptualizes this relationship between the Role As A State and the Role Domain Invariant. Finally, Figure 5 includes the Relator Obligation intended to represent the situation where one or many sub-goals of an abstract goal are domain properties. In this case these properties must hold in order to achieve such an abstract goal.

5.6.2 Responsibility model

Figure 1 shows two Relators intended to conceptualize the reification of the Role KAOSEnvironmentAgent and the Role KAOSRequirement Agent into respectively the stakeholder RoleMixin IoT Application Stakeholder (defined in Figure 2), and into the software agent RoleMixin KAOSIOTSoftwareAgent. This RoleMixin is a specialization of the Category Agent (defined in Figure 1).

5.7 Object and operation models

5.7.1 Object model

In the context of IoT applications, a KAOS object model represents relevant entities (passive objects) and agents belonging to a specific IoT domain and to an IoT application domain. These two domains, represented by their respective Kind (specialization of the Category Object), are modeled as components of the Kind KAOS Object Model.

5.7.2 Operation model

The operation model focuses on the satisfaction of the so-called operational goals (requirements and expectations), and more precisely on the means (operations) and ways to achieve them. KAOS operations are fully described by their trigger, their pre-conditions, their inputs, their outputs, and their eventual generation of events. All these operation components should be modeled. OntoUML provides a set of useful stereotypes intended for the conceptualization of situations (state-like concept), events (action and non-action events), and specific associations. We also reuse the UFO concepts of Atomic and Complex Action.

6 KAOS business aspect reinforcement

6.1 Preamble

Although business goals (high-level KAOS goals) are made explicit in the goal model, they are mainly used for the derivation of operational goals. KAOS models do not carry explicit information on business processes attached to

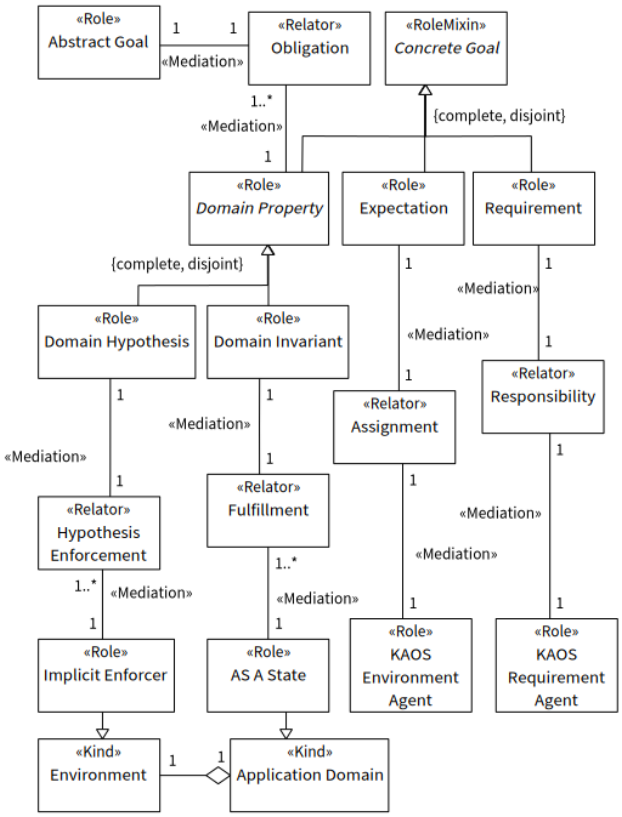


Figure 5: Concrete goal sorts

high-level business goals. The (native) KAOS framework did not give much consideration to the business dimension which is an important pillar for CIM modeling. Many works related to goal-oriented business modeling have been reported. Among these works, we mention GO4SoA [10], where the concept of business goal is incorporated to the SOA, a business architecture reputed centred around the concept of service. GO4SoA specifies SoA applications based on business goals. In this approach, as reported in [10], the goals become part of the semantic services' descriptions.

6.2 Proposal

We adopt another alternative which consists in reinforcing the business dimension of the KAOS modeling framework. This is achieved by augmenting our proposed KAOS ontology with a fragment intended to incorporate the capability modeling construct. We exploit the knowledge carried on one hand by the KAOS goal (top-level goals, abstract goals, concrete goals, and agents), and by the KAOS operations on the other hand. The main idea consists in identifying an additional KAOS-based construct "semantically" equivalent to the SoaML capability construct. This (added) construct, built from a set of specific (native) KAOS constructs is then used to set up an (indirect) mapping between KAOS and SoaML concepts. Firstly, top-level goals (abstract goals positioned just at the first level under the tree root), extracted from the KAOS goal model and representing strategic business goals, are considered as first class elements. They are called Top Business Goals. The last descendant goals (concrete goals) reachable from a given top-level goal constitute the set of operational goals to be achieved in order to satisfy the given top-goal. Secondly, we introduce the notion of KAOS Agent Capability. This notion is intended to represent the set of all KAOS atomic operations that an agent is capable to perform in order to achieve each of the concrete goals under its responsibility (one KAOS operation for each concrete goal). Thirdly, we extend and generalize the notion of KAOS Agent Capability representing the set of capabilities required from a group of KAOS agents to achieve a top-level KAOS goal: it is called Group-Agent Capability.

6.3 Augmented KAOS modeling

Figure 6 depicts a piece of ontology intended to conceptualize the new introduced concepts (Role Top Business Goal, Mode KAOS Agent Capability, and Mode KAOS Group Agent Capability), and their relationships with native KAOS concepts (Agent, Goal, and Operation). The diagram should be read as follows.

- The Role Top Business Goal is modeled as a specialization of the RoleMixin Abstract Goal (defined in Figure 4). The association named Leads to is intended to model that from a top business goal (Role Top Business Goal) a set of concrete goals (Role Concrete Goal) are reachable.

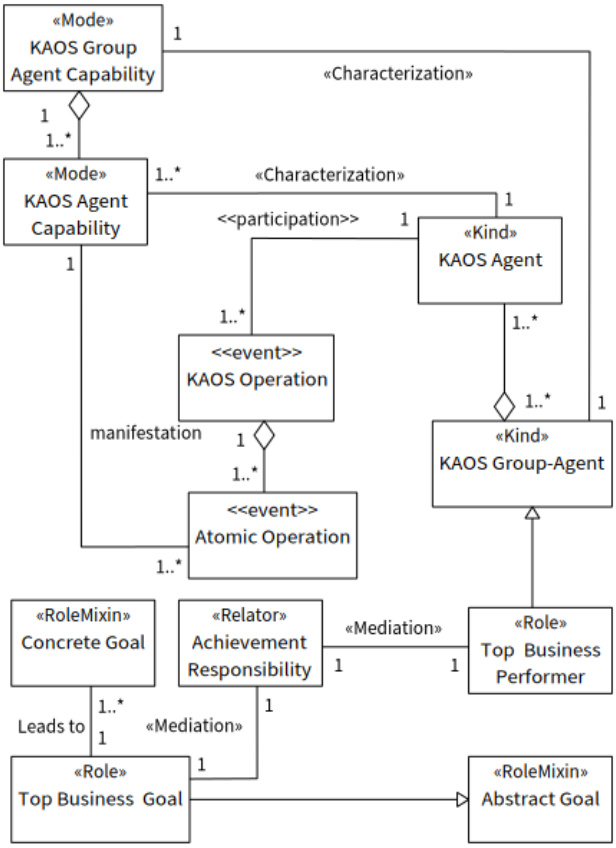


Figure 6: KAOS capabilities ontology

- The Relator Achievement Responsibility is intended to relate a top business goal (Role Top Business Goal) with its corresponding group of KAOS agents (Role Top Business Performer) which are together responsible for the top business goal achievement. These groups, named KAOS Group-Agent, are modeled as a composition of KAOS agents (Kind KAOS Agent).
- A KAOS Agent is responsible for a set of KAOS Operations. We define a KAOS Agent Capability as a coherent subset of the set of atomic operations composing a given KAOS operation; the choice of these subsets and their consistency is the responsibility of the relevant requirements' engineer. In this way, a KAOS Agent owns a set of capabilities. From the OntoUML perspective, KAOS Atomic Operations (considered as events) are a manifestation of the Capabilities of KAOS Agent.
- KAOS Agent Capability is modeled as a Mode class depicting a feature of a KAOS Agent. A stereotyped association (<<Characterization>>) links the Kind KAOS Agent to the Mode KAOS Agent Capability.
- A KAOS Group Agent Capability is characterized by

a feature intended to represent operational skills that it can potentially perform. This feature is modeled as a Mode class. Concretely it is modeled as a composition of the capabilities associated with all KAOS agents composing the group.

– The Kind KAOS Agent can participate into several KAOS operations. This is modeled by the association stereotyped by <<participation>>. KAOS operations are composed of a set of atomic operations (Atomic Operation).

7 Our preliminary SoaML ontology

7.1 Approach

The scope of this (part of) work encompasses the core SoaML business concepts and constructs. We are interested only in the constructs related to the modeling of the business aspect (CIM model). More precisely our SoaML ontology framework includes entirely the models of service, capability, participant, port, service interface, service architecture, and service contract. Our proposal builds on and refines specific UFO ontological elements to conceptualize concepts relevant to the SoaML business models, according to the SoaML universe of discourse. The SoaML metamodel as well as published works aiming at developing ontologies for the Service-oriented Architecture (SOA) [37, 28] are used as primary sources to capture this universe of discourse, augmenting by the way the degree of validity of our ontology fragment.

7.2 SoaML capability

A capability represents the ability of a SoaML entity to produce an outcome (business value) through a service. A SoaML service is a mechanism allowing to access exposed capabilities through an interface. Figure 7 depicts a piece of ontology conceptualizing the notion of SoaML capability and its relationship with the notion of a SoaML service. The diagram includes the Kind Service, a specialization of the Category Business Object, which is itself a refinement the Category Social Object. The Role Enabled Service, a specialization of the Kind Service, conceptualizes the fact that when services are created, a stereotyped association (<<Characterization>>) links the Category Business Object to the Mode Value, expressing the fact that business objects intrinsically carry values. The diagram shows two Relators (Production and Exposure): the Relator Production links the Mode SoaML Capability to the Mode Value, expressing the fact that a capability produces a value. The Relator Exposure links the Role Enabled Service, a specialization of the Kind Service, to the Mode SoaML Capability.The Relator Exposure conceptualizes the fact that when services are enabled, their capabilities become exposed to the environment.

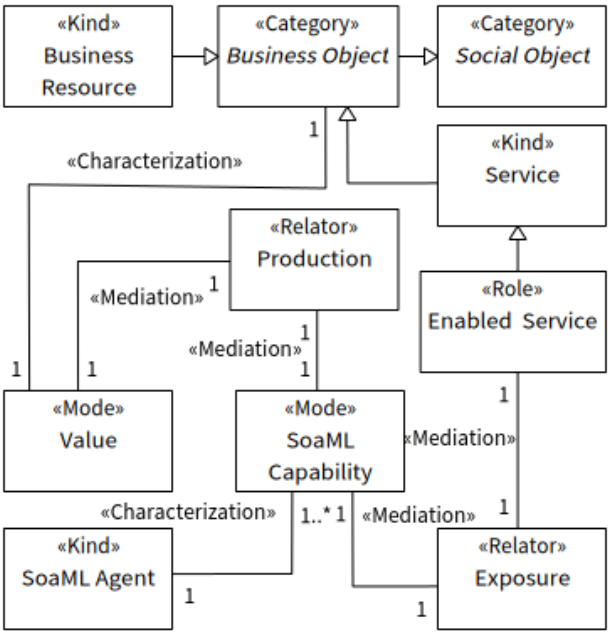


Figure 7: SoaML capability and service ontology

7.3 SoaML agent

According to the SoaML specification document [33], participants are constructs intended to represent domain entities able to provide and/or consume services through ports. In the business domain, participants may be business workers and stakeholders, corporation units, and artefacts like business architectures. In the computing domain participants may be software agents, and artefacts like software components and architectures. Two sorts of domain entities are identified in the SoaML metamodel: intentional entities (agents) and unintentional ones. These domain entities are abstracted by the concept of participants. (SoaML) agents are seen as a special sort of participants. (SoaML) ports are a kind of interaction points "anchored" to participants. SoaML identifies two sorts of ports, named respectively Request Port and Service Port: the first sort is used by service consumers to submit their requests to service providers, whilst the second one is used by service providers to offer their services. Figure 8 depicts a piece of ontology defining the concept of SoaML agent. The diagram in Figure 8 should be read as follows. SoaML Agent is conceptualized as a Kind that specializes the Category Agent. It is characterized by two features: the Mode Business Capability and the Mode Communication Capability. The figure also includes the Kind KAOS Agent, specializing the Category Artificial Agent and characterized by the Mode Communication Capability such as the SoaML Agent.

Figure 9 depicts a piece of ontology conceptualizing the participant, the ports' constructs as well as their interrelationship.

The depicted diagram should be read as follows. The

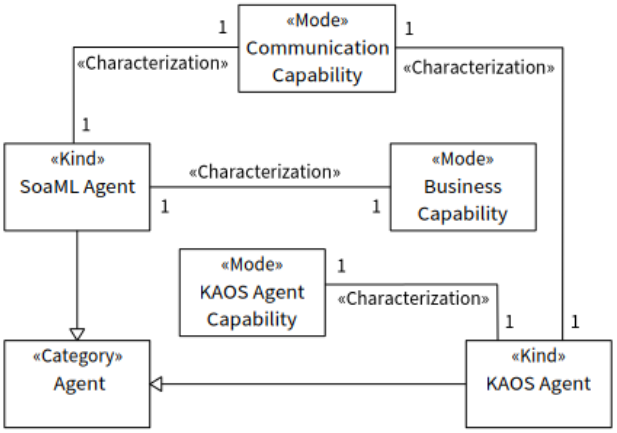


Figure 8: SoaML capability

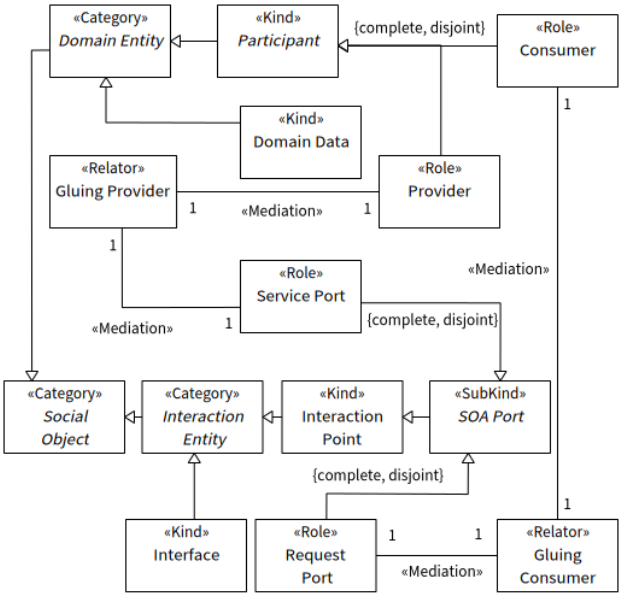


Figure 9: SoaML participant-port ontology

Category Social Object is specialized by two Categories. The first one named Domain Entity is intended to conceptualize the business entities belonging to domains. The second one named Interaction Entity is intended to model all sorts of interaction entities. The Kind Participant, specializing the Category Domain Entity, conceptualizes the concept of SoaML participant. The Role Provider and the Role Consumer, specializing the Kind Participant, conceptualize the fact that SoaML participants can play either the role of service provider, the role of service consumer, or both. We introduce a new Category, named Interaction Entity, intended to conceptualize various kinds of abstract interaction items such as interaction points, interaction protocols, interfaces, and connectors. The Kind Interaction Point, specializing the Category Interaction Entity, represents one sort of interaction entities. The Subkind SoA Port, a specialization of the Kind Interaction Point, conceptualizes the SoA

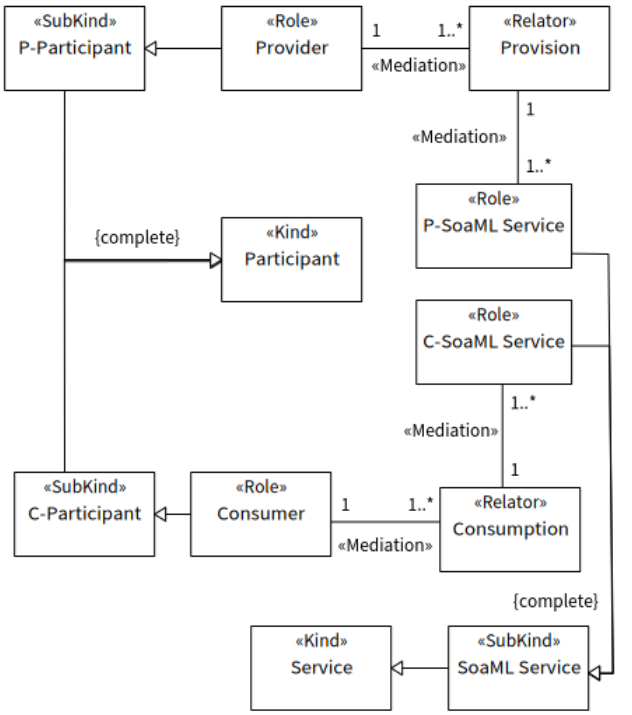


Figure 10: SoaML participant-service ontology

notion of port. There are two sorts of ports: the Request Port that represents the interaction point through which the service is requested, and the Service Port that represents the interaction point through which the service is offered. The SoA Port is specialized into the Role Service Port and the Role Request Port, thus emphasizing its specific roles, namely Request and Service. Two Relators, named Gluing Provider and Gluing Consumer, conceptualize the "glue" relationships respectively between Role Provider and Role Service ports, and between Role Consumer and Role Request ports.

7.4 SoaML participant and service

Figure 10 is intended to conceptualize the relationship between the participant and the service concepts. The Kind Participant is specialized by two Subkinds, namely P-Participant and C-Participant. They are dummy classes introduced to make the intended model more readable and semantically correct. P-Participant is specialized by the Role Provider, while C-Participant is refined into the Role Consumer.

The SoaML Service is modeled as a SubKind of the Kind Service. It is specialized by two Roles (P-SoaML Service, and C-SoaML Service). The Relators Provision and Consumption conceptualize the fact that SoaML services are offered or consumed by SoaML participants.

7.5 SoaML interfaces

As reported in [34], "service interfaces are used to describe provided and required operations to complete services' functionality". SoaML provides three sorts of interfaces:

- Simple Interfaces that give access to very basic services requiring a "one-way" interaction (request/response pattern), where a service consumer calls the (unique) operation offered by service provider.
- Service Interfaces that give access to services requiring bidirectional interactions (conversation pattern) between a service provider and a service consumer. A service interface types a service port belonging to the provider. From the perspective of a service provider, three elements completely define Service Interfaces: the interface it realizes and those it uses in order to offer and accomplish its provided service, the description of the (required) conversation between the service provider and its consumer, and an enclosed part specifying the specific role played by each of the connected participants. This last element is not addressed in the current work.

Figure 11 depicts a diagram intended to conceptualize the SoaML interface notion. The diagram includes also useful and required surrounding modeling elements. The modeling elements depicted in this figure are briefly described in the following.

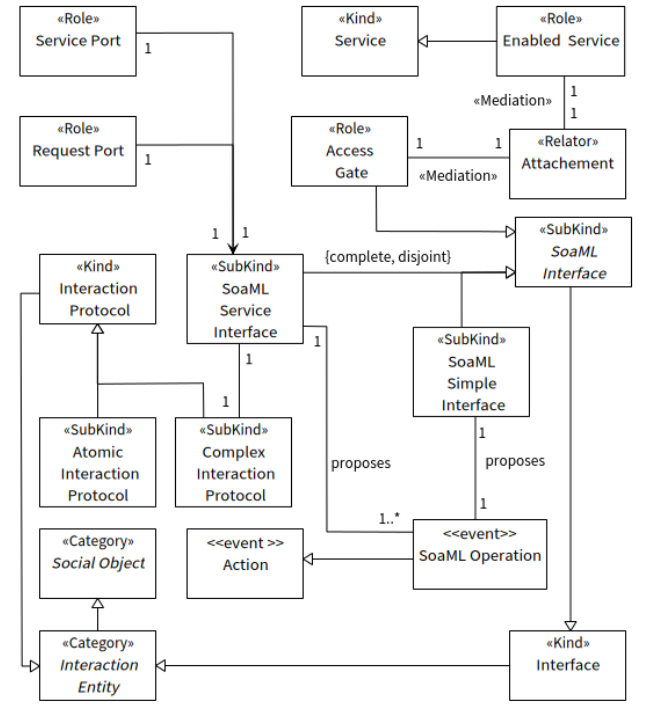


Figure 11: SoaML interfaces ontology

- Surrounding Modeling Elements
  We define the Category Interaction Entity as a specialization of the Category Social Object. Two Kinds, specializing the Category Interaction Entity, conceptualize respectively the general concepts of Interface and Interaction Protocol. On the one hand, we define SoaML Interface as a special sort of Interface, and on the other hand, two sorts of Interaction Protocol, namely Atomic Interaction Protocol intended to represent a simple one-way interaction protocol and Complex Interaction Protocol for bidirectional interactions. Conversations (Complex Interaction protocols) are modeled as ordered sequences of atomic interactions.
- Interfacing
  SoaML advocates the separation between the interface giving access to a service (visible part) and the service implementation (hidden part). The Role Access Gate, a specialization of the SubKind SoaML Interface, represents the visible part of the SoaML service. The Role Enabled Service, a specialization of the Kind Service, represents the service hidden part.
- SoaML Interfaces
  The subkinds SoaML Simple Interface and SoaML Service Interface, specializing the Kind SoaML Interface, conceptualize the SoaML simple interface and the SoaML service interfaces. The association named proposes expresses the fact that a simple interface offers a unique SoaML operation modeled as <<event>> Action. an event Action is intended to represent operations, tasks, and processes [21]).
- SoaML Service Interface
  The SoaML Specification Document [33] specifies two typing constraints: "the Service Interface is the type of a "Service" port on a provider and the type of a "Request" port on the consumer".

7.6 Services' architecture

This sub-section focuses on an important SoaML construct, namely Services' Architecture. It represents a high-level SoaML business construct that describes how participants work together for a purpose by providing and using services expressed as service contracts [33]. They are mainly used to define and specify compound services resulting from a composition of services provided and consumed by a community of (two or more) participants. A Services' architecture consists of a diagram linking services (instances) to be composed, with both their corresponding providers (instance) and consumers (instance). It can be derived from appropriate instantiations of the piece of ontology depicted in Figure 10. Because OntoUML does not provide means to instantiate instances (UFO individuals), we cannot express service architecture with OntoUML diagrams. However, we can visualize a services' architecture as a set of related

pairs, the first element of such pairs representing an instantiation of the Relator Provision and its second element representing an instantiation the Relator Consumption.

8 Merging KAOS and SoaML ontologies

The merging operation between two source ontologies is generally done through their potential shared (or overlapped) native concepts (if any), and/or through the elaboration of appropriate ontological bridges (pieces of intermediate ontologies). In this work we use both approaches in a complementary way. The first approach requires a prior identification and deep analysis of such shared native concepts, whilst the second requires a prior identification of suitable anchoring points, in both source ontologies, where the bridge may be attached. The ontology mapping operation is considered as a preliminary step along a process of merging or integration of ontologies. It consists in the precise characterization of the semantic correspondence (semantic similarity) relating potentially two or more "similar" concepts. Such correspondences between concepts may be simple (one to one), or complex (via a transforming operation or a logical expression relating similar concepts). In some situations, an appropriate intermediate piece of ontology is used as a bridge between similar concepts of different source ontologies. The analysis of both KAOS and SoaML modeling languages reveals two sorts of situations: (1) both languages share in an explicit way similar concepts (KAOS domain object versus SoaML participant; KAOS agent versus SoaML agent; KAOS operation versus SoaML operation); (2) one of these languages provides a concept in an explicit way, while a similar concept is hidden or implicit in the other language (KAOS goal versus SoAML business objective; KAOS capability versus SoaML capability).

– Explicit Similar Concepts

  (a) Domain Object versus Participant
  Both concepts refer to intentional (active) and non-intentional (passive) entities. However there are two slight differences: Participants provide /consume services, whilst the concept of service is not an inherent feature of domain objects; participants communicate through message exchanges, whilst the communication aspect between domain objects is left unspecified. Although these two concepts are not semantically equivalent, we can say that they are sibling concepts, because of their overlapping correspondence. One way to put these two sibling concepts together is to root them to a common parent concept. In our ontology, the Object category fills this need.

  (b) KAOS Agent versus SoaML Agent
  A priori, they are similar in the sense that both are intentional entities, but KAOS agents can be refined to distinguish two specific agents, which is not the case for SoaML agents. Although these two concepts are not semantically equivalent, we can say that they are sibling concepts because of their overlapping correspondence. This situation is treated in the same way as in the previous case.

  (c) KAOS Operation versus SoaML Operation
  At first sight these concepts seem to be very similar. However there are two differences between them: in contrast to KAOS operations, which generally are complex actions intended to satisfy requirements, SoaML operations are atomic actions, offered through specific interfaces, and intended to provide services; SoaML operations can either be called (request/response) or invoked through message exchanges, while KAOS operations are triggered by events. It is worth noting that message sending and receiving are themselves sorts of non-action events.

– Implicit Similar Constructs

  (a) SoaML Capability versus KAOS Capability
  SoaML provides explicit constructs intended to represent both the capability and service concept, the service concept being built on the capability concept. The capability concept, though mentioned in the reported KAOS literature, is not perceived as an explicit KAOS construct. In this situation, our approach consists in extending KAOS in such a way to make explicit the notion of KAOS capability, and to define it in such a way to make possible a mapping to the SoaML similar concept. Furthermore, the proposed extension ensures, through this mapping, a "natural connection" with the notion of service.

  (b) KAOS Goal versus SoaML Business Objective
  KAOS deals in an explicit way with goals distinguishing between business (top level) and operational goals (leaf level) thanks to their relative positions in the so-called KAOS goal model. Regarding SoaML, it does not support an explicit goal construct. However, business goals and processes are supposed to be (previously) identified in an explicit way using an appropriate business modeling language, and then SoaML models (capabilities, service contracts, and services' architecture) may be derived from these identified business goals and processes. Business goals are considered only as implicit input knowledge exploited by SoaML modelers. Thus we may conclude that KAOS goals have no similar SoaML construct.

9 Instantiating OntoUML models

9.1 Transformation approach

The OntoUML models describing our proposed ontology, once designed, edited and their syntax and semantics verified, are ready to be instantiated in concrete OWL classes and populated with concrete individuals (lowest instance level) related to specific IoT applications, thus leading to a "query-able" Knowledge/Data graph. More precisely, OntoUML models can be transformed (manually or automatically) into appropriate gentle UFO (gUFO) classes. gUFO is an extension of OWL supporting UFO theoretical framework as well as the set of OntoUML stereotypes. To this end, we use the following approach. First, we transform (a selected subset of) our proposed ontology fragments into their corresponding gUFO descriptions according to UFO semantic rules. Second, we instantiate these gUFO descriptions into specialized gUFO classes and concrete individuals corresponding to the needs expressed by stakeholders (cf. Section 10).

The transformation is performed as follows. Our OntoUML (stereotyped) classes (reflecting a reality such as KAOS, and generic IoT concepts independent from a specific IoT application) are directly transformed into corresponding gUFO classes, i.e., gUFO:Category, gUFO:Kind, gUFO:SubKind, gUFO:Role, gUFO:RoleMixin, gUFO:Relator, and others. This is performed, of course, according to the class hierarchies and relationships belonging to our OntoUML models. For this purpose we mainly use rdf and rdfs sentences such as
:C rdf:type gufo:X.
:C rdfs:subClassOf gufo:D.
:C rdfs: superClassOf gufo:E.
The first sentence means that C, a stereotyped class belonging to our model, is a class instance of the gUFO class X, the second sentence means that C is a subclass of D, and the third sentence means that D is a superclass of E. These descriptions are useful to correctly describe the hierarchy of the stereotyped classes belonging to our OntoUML ontology fragments. Other specific rdf sentences are used to express mediations linking OntoUML classes and Relators.

9.2 Samples of queries on our populated ontology fragments

This subsection presents samples of generic queries that may be issued by relevant users to the ontology fragments populated with instances related to IoT applications under specification. The intended users of these ontology fragments are the stakeholders engaged in the CIM development of IoT applications. These samples are categorized as follows.

Goal Hierarchy

1.1 What are the top goals? 1.2 What are the concrete goals associated with a top goal? 1.3 What are the children goals of a parent goal? 1.4 Which kind of composable goal is a parent goal?

Goal-Agent-Stakeholder-Operation

2.1 List the requirement agents. 2.2 Which stakeholder is responsible for a goal? 2.3 What are the domain properties enforced by an environment agent? 2.4 What are the domain properties required by an abstract goal? 2.5 Which expectation is as- signed to an environment agent? 2.6 Which requirement agent is responsible for a requirement? 2.7 List the expectations. 2.8 List the domain invariants 2.9 Which stakeholder is reified by an environment agent? 2.10 Which software agent is reified by a requirement agent? 2.11 List the input parameters of an operation. 2.12 List the output parameters of an operation. 2.13 Which agent participates to an operation? 2.14 What are the capabilities of an agent?

IoT

3.1 List the IoT sensors 3.2 List the IoT actuators 3.3 List the software agents 3.4 Which software agent is embedded into a IoT device? 3.5 Which software agent is hosted in a computing device ? 4.1 List the participants. 4.2 List the agents. 4.3 What is the business capability of a participant? 4.4 What is the communication capability of a participant? 4.5 List the services. 4.6 What are the capabilities exposed by a service? 4.7 What is the value produced by a capability? 4.8 What is the interface of a service? 4.9 What is the provider participant of a service interface? 4.10 What are the ports of a service interface? 4.11 What is the operation provided by a simple interface?

10 An illustrative example

This section has two parts. A problem statement for our illustrative IoT application, along with the needs elicited, is first outlined, then followed by a gUFO instantiation of selected OntoUML models represented in Figures 1, 2, 3 , 4, and 5.

10.1 Problem statement and needs' elicitation

10.1.1 Problem statement

The chosen example scenario, inspired from the one described in [11], is adapted and enriched in order to fulfil our illustrative needs. A platform, equipped with various smart sensors and actuators located inside the patient home, is intended to monitor and control a set of patient vital parameters, of home-related parameters (temperature, humidity), and of security-related parameters (face recognition). The objective of this platform is to maintain patient health condition and to provide patients with medical care in case of emergency situations.

10.1.2 Elicitation

(a) Stakeholders
Corporation owning and managing the platform: Manager, Remote Patient Followers, Health CareGiving Centers, Ambulance Supplying Centers, Security Centers, Patients.

(b) Domain Entities

+ IoT devices: smart sensors (medical, temperature, humidity, face recognition camera, windows' status monitor), smart actuators (dehumidifier, air conditioner, window opener/closer).
+ Health - vital parameters: heartbeat, blood pressure, oxygen rate, and so on.
+ Home Infrastructure Hypothesis: automatic (opening/closing) windows, Internet/IoT infrastructure, Health Platform installed.

(c) Goal Model

+ Functional Goals
  – Root goal: Manage Patient.
  – First-level goals (AND): Keep Patient Healthy, Achieve Patient Authentication.
  – Keep Patient Healthy (AND): Patient Condition Monitoring, Corrective CounterMeasures.
  – Patient Condition Monitoring (AND): Medical Parameters Monitoring, Home Parameters Monitoring.
  – Medical Parameters Monitoring (AND): Heartbeat Measurement, Blood Pressure Measurement, Oxygen Rate Measurement.
  – Home Parameters Measurement(AND): Temperature Measurement, Humidity Rate Measurement.
  – Corrective CounterMeasures (OR): Medical Care CounterMeasures, Home CounterMeasures.
  – Home CounterMeasures (AND): Control Temperature, Control Humidity. Control Temperature (OR): Device based Air Conditioning, Natural Air Conditioning.
  – Control Humidity (OR): Device Based De-Humidification, Natural De-Humidification.
  – Medical Care CounterMeasures (OR): Emergency Evacuation, At Home CareGiving.
  – Achieve Patient Authentication (AND): Patient Registration, Patient Authentication Checking.
  – Patient Registration (AND): Platform Prompting, Patient Information Filling.
  – Patient Authentication Checking(AND): Face Capturing, Face Checking.
+ Domain Properties
  – Domain Invariants: Admitted ranges of vital and home parameters values.
  – Domain Expectations: Specific Home Appliances installed and working, IoT Internet available.
+ Listing and categorizing of the potential pertinent agents
  – Users and Workers: Patient (IoT Application User), HomePlatform Product Responsible (manager), At-Home CareGiver (business worker), Remote Patient Follower (business worker), Security Checker (business worker).
  – Corporation Units: Health CareGiving centers, Ambulance supplying centers, Security centers.
  – Software
    · Application Specific Software: Patient Platform Manager.
    · IoT Software: A specific Software Agent for each smart sensor and actuator engaged in the platform.

10.2 Transformation into gUFO and instantiations

In this section we apply the two-steps' approach stated in Section 9.

10.2.1 Transformation into gUFO

We cover almost all sorts of stereotyped classes used in our OntoUML models. The following shows the transformation of a set of significant classes belonging to some selected figures.

+ **KAOS Agent Ontology** (cf. Figure 1) **Category and Kind** :Agent rdf:type gufo:Category.
:KAOSAgent rdf:type gufo:Kind;
rdfs:subClassOf :Agent.
Role and RoleMixin
:KAOSEnvironmentAgent rdf:type gufo:Role;
rdfs:subClassOf :KAOSAgent.
:KAOSIoTSoftwareAgent rdf:type gufo:RoleMixin;
rdfs:subClassOf :Agent;
rdfs:superClassOf :IoTDeviceSoftwareAgent;
rdfs:superClassOf
:ApplicationSpecificSoftwareAgent.

+ **IoT Software Ontology** (cf. Figure 3)
RoleMixin and Role

:KAOSIoTSoftwareAgent rdf:type gufo:RoleMixin;
rdfs:subClassOf :Agent;
rdfs:superClassOf :IoTDeviceSoftwareAgent;
rdfs:superClassOf
:ApplicationSpecificSoftwareAgent.
:IoTSensorSoftwareAgent rdf:type gufo:Role;
rdfs:subClassOf :IoTDeviceSoftwareAgent.
:IoTActuatorSoftwareAgent rdf:type gufo:Role;
rdfs:subClassOf :IoTDeviceSoftwareAgent.

+ **KAOS Goal Hierarchy** (cf. Figure 4)
Category and RoleMixin
:Goal rdf:type gufo:Category.
:RequirementEngineeringGoal rdf:type
gufo:Category;
rdfs:subClassOf :Goal.
:KAOSGoal rdf:type gufo:RoleMixin;
rdfs:subClassOf :RequirementEngineeringGoal.

+ **Role and RoleMixin**
:RetainedKAOSGoal rdf:type gufo:RoleMixin;
rdfs:subClassOf :KAOSGoal;
:rdfs:superClassOf :ConcreteGoal;
rdfs:superClassOf :AbstractGoal.
:AbstractGoal rdf:type gufo:RoleMixin;
rdfs:subClassOf :RetainedKAOSGoal;
rdfs:superClassOf :ORDecomposable;
rdfs:superClassOf :ANDDecomposable.
:ORDecomposable rdf:type gufo:Role;
rdfs:subClassOf :AbstractGoal.
:ANDDecomposable rdf:type gufo:Role;
rdfs:subClassOf :AbstractGoal.

+ **Concrete Goal Sorts Ontology** (cf. Figure 5)
Role and RoleMixin
:ConcreteGoal rdf:type gufo:RoleMixin;
rdfs:subClassOf :RetainedKAOSGoal;
rdfs:superClassOf :Requirement;
rdfs:superClassOf :Expectation;
rdfs: superClassOf :DomainProperty.
:Requirement rdf:type gufo:Role;
rdfs:subClassOf :ConcreteGoal.
:Expectation rdf:type gufo:Role;
rdfs:subClassOf :ConcreteGoal.
:DomainProperty rdf:type gufo:Role;
rdfs:subClassOf :ConcreteGoal.
:DomainHypothesis rdf:type gufo:Role;
rdfs:subClassOf :DomainProperty.
Relator and Mediation
:Assignment rdf:type gufo:Relator.
:AssignmentInvolves rdf:type owl:ObjectProperty;
rdfs:subPropertyOf gufo:mediate;
rdf:domain :Assignment;
rdf:range :Expectation;
rdf:range :KAOSEnvironmentAgent.

10.2.2 Instantiations of concrete individuals

The following shows samples of concrete individual instantiations. These are either directly instantiated from stereotyped classes of our OntoUML models or in an indirect way through the specialization of stereotyped classes of our OntoUML models.

(A) Direct concrete individuals instantiations

+ **Abstract goals** (samples)
Patient (:ag0), Keep Patient Healthy (:ag1), Achieve Patient Authentication (:ag2), Patient Condition Monitoring (:ag3).
:ag0 rdf:type :AbstractGoal, …

+ **Concrete goals** (samples)
Blood Pressure Measurement (crg2), Oxygen Rate Measurement (crg3)
Temperature Measurement (crg4) are instances of :Requirement Role
:crg2 rdf:type :Requirement
Patient Information Filling(ceg13) is an instance of :Expectation Role
:ceg13 rdf:type :Expectation

+ **IoT Devices** (samples)
dhd: rdf:type IoTActuator; (dehumidifier device)
:wd rdf:type IoTActuator; (window actuator device)
:tempd rdf:type :IoTSensor; (temperature device)
:hbd rdf:type :IoTSensor. (heartbeat device)

+ **Environment Agent**
Only two instances can be instantiated from this Role class
:asanappuser rdf:type :KAOSEnvironmentAgent;
:asacorpworker rdf:type :KAOSEnvironmentAgent.

(B) Indirect concrete individual instantiations

+ **Patients**: First we specialize the Role class IoT Application User (cf. Figure 2) into the class Role Patient, then we instantiate concrete individuals of the class Patient
:Patient rdf:type gufo:Role
rdfs:subClassOf IoTApplicationUser
:patient1 rdf :Patient

+ **Stakeholder**: First we specialize the Role class IoT Application Corporation Stakeholder (cf. Figure 2) into four Role classes :AtHomeCareGiver, RemotePatientFollower, SecurityChecker, and Home-PlatformProductResponsible.

Then we instantiate concrete individuals of these four Role Classes:
:atg rdf:type :AtHomeCareGiver;
:rpf rdf:type :RemotePatientFollower;
:sc rdf:type :SecurityChecker;
:hppr rdf:type :HomePlatformProductResponsible.

11 Concluding remarks, discussion, and future research directions

11.1 Results

This work presents an ontological-based framework intended to help CIM modelers in their preliminary analysis of IoT applications under development. The use of ontologies is mainly motivated by their high aptitude to address descriptive aspects inherent to CIM modeling, and the profusion of available domains' ontologies making them reusable in various IoT applications. The proposed framework relies on a combination of KAOS (a good candidate for the requirements and domain modeling) and SoaML (for the business service modeling). The outcomes of this work are fourfold. Preliminary ontologies for KAOS and SoaML, addressing their respective key concepts, as well as basic IoT elements, are built; a pragmatic merging of these two ontologies is proposed; an implementation of our ontology fragments in gUFO classes aimed at their computerization is given; an illustrative example demonstrating the applicability of our results to IoT applications is presented. It is worth mentioning, that OpenPonk [8] was used to edit and verify the syntax and the semantics of our models before transforming them into gUFO. The semantics is checked against well-defined anti-patterns.

11.2 Discussion

In this section, we first focus the discussion around the comparison of our work with (only) those mentioned in Table 1, i.e., the works that are the closest to our work; then we compare our proposed KAOS and SoaML ontologies with those reported in the literature.

11.2.1 Comparison with works reported in Table 1

According to Table 1, [38] is (to the best of our knowledge) the sole work combining KAOS and SoaML concepts aimed at addressing IoT requirements. What primarily distinguishes our work from[38] is our adoption of ontological metamodeling rather than "traditional" UML metamodeling. The advantage of this alternative approach reported in [24] is based on the mapping of elements of modeling languages to appropriate ontological concepts; it lies mainly in the improvement of the semantic perspective thanks to the use of well-defined and sound (OntoUML) stereotypes. Our ontology fragments are built according to top-level UFO goal and agent fragments on one side, and KAOS and SoaML metamodels on the other side. Additionally, the resulting OntoUML models, as conceptual ontologies, are transformed into either operational ontologies (gUFO/OWL) or formal specifications (Alloy formal specification language) for further analysis and simulation purposes. [43] enriched KAOS/SysML (a combination of KAOS and SysML) with an ontological approach. However, unlike our entirely ontological approach, the concept of ontology used in [43] covers only the domain aspect. Furthermore, our proposal, aimed at IoT applications, not only explicitly addresses KAOS software agents specific to the IoT context, but also addresses the application domain (KAOS Object Model) from an IoT perspective. [11] provided a UFO ontological approach bringing together the concepts of Goal, Agent, Task, and Service. Their objective differs from ours: first, their proposed ontology serves as a "framework" for developing metamodels for new domain-specific modeling languages, while ours reinforces the semantics of metamodels of an existing, well-established modeling language (KAOS and SoaML); secondly, they consider the SoA principle, where services are traditionally discovered and searched from the outside of the "calling" application, while on our side we focus on the SoaML modeling language rather than the mentioned SoA principle. The novelty consists in an ontological bridging between our proposed KAOS ontology fragments and SoaML fragments: bringing "semantically" closer KAOS agents and SoaML participants, KAOS Goal and SoaML Service through their "shared" capability concept which is implicitly mentioned in the KAOS literature and explicitly defined in the SoaML metamodel. This way facilitates a shift from a KAOS goal model to a SoaML architecture. Finally, [13] proposes a goal oriented methodology intended for AAL requirements (GoAAL), that is mixing ontology fragments from diverse sources (such as IoT domain and health domain). The work adopts a goal ontology fragment based on a variant of i-star, rather than KAOS, for goals. Although the concepts of task and operation are retained, the service concept is not considered in [13].

11.2.2 KAOS and SoaML ontologies

The literature reported ontology proposal(s) for KAOS [27], and for SoaML [30, 37, 29]. In [27], the authors used on one hand a standard KAOS metamodel as a reference to build their KAOS ontology, and on the other hand the Unified Enterprise Modeling Language (UEML) as an ontological modeling language. UEML is usually intended for enterprises and information systems' modeling. Our KAOS ontology not only refers to a standard KAOS metamodel, but is also based on proposed refinements of modeling elements drawn from UFO-C (Agent and Goal) ontology fragments [19, 23], and also on proposed refinements of the Goal-oriented Requirement Ontology (GORO) [31, 23]. Regarding SoaML, [30] and [37, 29] are among the recent works aiming at introducing ontology concepts for SoA and SoaML. The authors of [30] have analyzed and evaluated

the use of UFO-S, a UFO sub-ontology for services, in various approaches including the SoaML one; those of [37] have proposed a set of ontologies covering the general aspects of the so-called Service Engineering (service-oriented architecture, software service ontology, etc.) including ontologies for SoaML. Their work relies on the Open Group Service Ontology, and also on the ISO/IEC SoA Reference Architecture. [29] provides a comparison between the UFO-S service ontology and other similar service ontologies including the Open Group Service Ontology. Our SoaML ontology is built using refinements and adaptations of some modeling elements of UFO-A and UFO-B ontology fragments. The suggested adaptations mainly rely on relevant knowledge drawn from the OMG SoaML specification document [33] as well as from [37, 29].

11.3 Limitations and future research directions

This work consists in proposing an ontological conceptualization of IoT CIMs initially modeled using a combination of KAOS and SoaML, with KAOS addressing both requirements and domain aspects, while SoaML addressing the business aspect through the service and service architecture concepts. A set of preliminary related ontology fragments illustrated by samples of instantiations constitutes our main results. At this first and current stage of our work, we identified some limitations that deserve to be addressed in future contributions. A first limitation concerns the use of structural models in general and OntoUML in particular to address business process modeling. We envisage for a future work to investigate a type of model more adapted to process modeling such as BPMN (Business Process Model and Notation) or a combination of BPMN and DEMO (Design & Engineering Methodology for Organisations). A second limitation comes from our (re)-use of the Agent and Goal UFO-C fragments, while for the time being gUFO does not support UFO-C, which prevents their potential instantiation using gUFO. In order to make our fragments more reliable, an extension of gUFO constitutes a future research direction. In our proposed fragments, we introduced some aspects related to the IoT world, particularly IoT Software Agents and IoT devices. We also suggested that, in the context of IoT, the KAOS Object Model encloses available public ontologies related to the IoT domain as well as the application domains. However, we adopted in our approach a generic and general purpose SoaML service concept. A future development of our work in this direction using a combination of BPMN and DEMO seems to be a promising future work.
As other future works, we plan to:

- transform OntoUML based ontologies into Alloy formal specifications for verification and simulation purposes [7].
- Enrich the KAOS ontology by adding the conceptualization of the obstacle and conflict concept.

Acknowledgement

The authors thank the anonymous reviewers for their valuable comments that helped improve this version of the paper.

References

[1] S. Assar. Model driven requirements engineering mapping the field and beyond. In *Model Driven Requirement Engineering Workshop MoDRE*, 2014. https://dx.doi.org/10.1109/MoDRE.2014.6890820.

[2] U. Aßmann, S. Zschaler, and G. Wagner. Ontologies, meta-models and the model-driven paradigm. In *Ontologies for Software Engineering and Software Technology*. Springer Berlin Heidelberg, 2010. https://dx.doi.org/10.1007/3-540-34518-3_9.

[3] S. Benkhaled, M. Hemam, M. Djezzar, and M. Maimour. An ontology – based contextual approach for cross-domain applications in Internet of Things. *Informatica An International Journal of Computing and Informatics*, 2022. https://doi.org/10.31449/inf.v46i5.3627.

[4] C. H. Bernabe, V. E. S. Souza, R. de Almeida Falbo, R. S. S. Guizzardi, and C. Silva. GORO 2.0: Evolving an ontology for goal-oriented requirements engineering. In *Advances in Conceptual Modeling ER*, 2019. https://dx.doi.org/10.1007/978-3-030-34146-6_15.

[5] M. Bettaz. Implementing OntoUML Models with OntoObject- Z Specifications: A Proof of Concept Relying on a Partial Ontology for VLANs. In *14th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2024*. SciTePRESS - Science and Technology Publications, Lda, 2024. https://dx.doi.org/10.5220/0012854500003758.

[6] M. Bettaz and M. Maouche. Towards a New Ontology-based Descriptive Language: OntoObject-Z. In *International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2023. https://dx.doi.org/10.1109/IC3I59117.2023.10397921.

[7] B. F. B. Braga, J. P. A. Almeida, G. Guizzardi, and A. B. Benevides. Transforming OntoUML into Alloy: towards conceptual model validation using a lightweight formal method. *Innovations in Systems and Software Engineering*, 2010. https://dx.doi.org/10.1007/s11334-009-0120-5.

[8] CCMI. OpenPonk platform. https://ccmi.fit.cvut.cz/tools/openponk/, 2023.

CCMI Research Group, Faculty of Information Technology, Czech Technical University in Prague.

[9] B. Costa, P. F. Pires, and F. C. Delicato. Modeling SOA-based IoT Applications with SoaML4IoT. In *World Forum on Internet of Things (WF-IoT)*, 2019. https://dx.doi.org/10.1109/WF-IoT.2019.8767218.

[10] I. C. Costa and J. M. P. de Oliveira. GO4SOA: Goal-oriented modeling for soa. In *International Conference on Web Information Systems and Technologies*, 2016. https://dx.doi.org/10.5220/0005800902470254.

[11] L. O. B. da Silva Santos, G. Guizzardi, and R. S. S. Guizzardi. GSO: Designing a well-founded service ontology to support dynamic service discovery and composition. In *Enterprise Distributed Object Computing (EDOC)*, 2009. https://dx.doi.org/10.1109/EDOCW.2009.5332016.

[12] D. Dermeval, J. Vilela, I. I. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva. Applications of ontologies in requirements engineering: a systematic review of the literature. *Requirements Engineering*, 2016. https://dx.doi.org/10.1007/s00766-015-0222-6.

[13] C. Diamantini, A. Freddi, S. Longhi, D. Potena, and E. Storti. A goal-oriented, ontology-based methodology to support the design of AAL environments. *Expert Systems With Applications*, 2016. https://dx.doi.org/10.1016/j.eswa.2016.07.032.

[14] B. Elvesæter, C. Carrez, P. Mohagheghi, A.-J. Berre, S. G. Johnsen, and A. Solberg. Model-driven service engineering with soaml. In *Service Engineering Book*. Springer, 2011. https://dx.doi.org/10.1007/978-3-7091-0415-6_2.

[15] S. J. T. Fotso, M. Frappier, R. Laleau, A. Mammar, and M. Leuschel. Formalisation of SysML/KAOS Goal assignments with b system component decompositions. In *Integrated Formal Methods (IFM)*, 2018. https://dx.doi.org/10.1007/978-3-7091-0415-6_210.1007/978-3-319-98938-9_22.

[16] X. Franch, L. López, C. Cares, and D. Colomer. The i* framework for goal-oriented modeling. In *Domain-Specific Conceptual Modeling: Concepts, Methods and Tools*. Springer, 2016. https://dx.doi.org/10.1007/978-3-319-39417-6_22.

[17] G. Giancarlo, B. B. Alessander, F. Claudenir, P. Daniele, A. J. Paulo, and P. S. Tiagoa. UFO: Unified foundational ontology. *Applied Ontology*, 2022. https://dx.doi.org/10.3233/AO-210256.

[18] N. Guarino. Formal ontologies and information systems. In *Formal Ontology in Information Systems (FOIS)*. IOS Press, 1998.

[19] G. Guizzardi, R. de Almeida Falbo, and R. Guizzardi. Grounding software domain ontologies in the unified foundational ontology (UFO): The case of the ODE software process ontology. In *Conferencia Iberoamericana de Software Engineering*, 2008.

[20] G. Guizzardi and G. Wagner. Using the unified foundational ontology (UFO) as a foundation for general conceptual modeling languages. In *Theory and Applications of Ontology: Computer Applications*. Springer, 2010. https://dx.doi.org/10.1007/978-90-481-8847-5_8.

[21] R. S. Guizzardi and G. Guizzardi. Applying the UFO ontology to design an agent-oriented engineering language. In *Conceptual Modeling ER*, 2013. https://doi.org/10.1007/978-3-642-15576-5_16.

[22] R. S. S. Guizzardi and G. Guizzardi. Applying the UFO ontology to design an agent-oriented engineering language. In *Advances in Databases and Information Systems (ADBIS)*, 2014. https://doi.org/10.1007/978-3-642-15576-5_16.

[23] R. S. S. Guizzardi, G. Guizzardi, A. Perini, and J. Mylopoulos. Towards an ontological account of agent-oriented goals. In *Software Engineering for Large-scale Multi-Agent Systems (SELMAS)*, 2006. https://dx.doi.org/10.1007/978-3-540-73131-3_9.

[24] K. Hinkelmann, E. Laurenzi, A. Martin, and B. Thönssen. Ontology-based metamodeling. In *Business Information Systems and Technology 4.0*. Springer, 2018. https://dx.doi.org/10.1007/978-3-319-74322-6_12.

[25] L. Kadakolmath and U. D. Ramu. Goal-oriented modeling of an urban subway control system using KAOS. *Indonesian Journal of Computer Science (IJCS)*, 2023. https://doi.org/10.33022/ijcs.v12i3.3239.

[26] D. Man. Ontologies in computer science. *DIDACTICA MATHEMATICA*, 31(1):43–46, 2013.

[27] R. Matulevicius, P. Heymans, and A. L. Opdahl. Ontological analysis of KAOS using separation of reference. In *Contemporary Issues in Database Design and Information Systems Development*. IGI Global, 2007. https://doi.org/10.4018/978-1-59904-289-3.ch002.

[28] J. C. Nardi, J. P. A. Almeida, P. H. A. da Silva, and G. Guizzardi. An ontology-based diagnosis of mainstream service modeling languages. In *International Enterprise Distributed Object Computing*

Conference (EDOC), 2019. https://doi.org/10.1109/EDOC.2019.00023.

[29] J. C. Nardi, R. de Almeida Falbo, J. P. A. Almeida, G. Guizzardi, L. F. Pires, M. J. van Sinderen, and N. Guarino. Towards a commitment-based reference ontology for services. In *Enterprise Distributed Object Computing (EDOC)*, 2013. https://doi.org/10.1109/EDOC.2013.28.

[30] J. C. Nardi, R. de Almeida Falbo, J. P. A. Almeida, G. Guizzardi, L. F. Pires, M. J. van Sinderena, N. Guarino, and C. M. Fonseca. A commitment-based reference ontology for services. *Information Systems*, 2015. https://doi.org/10.1016/j.is.2015.01.012.

[31] NEMO. Goal oriented requirements ontology (GORO). https://dev.nemo.inf.ufes.br/seon/GORO.html. Research Group.

[32] J. C. Nwokeji, T. Clark, and B. S. Barn. Towards a comprehensive meta-model for KAOS. In *Model-Driven Requirements Engineering (MoDRE)*, 2013. https://doi.org/10.1109/MoDRE.2013.6597261.

[33] OMG. Service oriented architecture modeling language (soaml) specification, v 1.0.1. https://www.omg.org/spec/SoaML/1.0.1/PDF. Object Management Group.

[34] M. A. Orellana, J. R. Silva, and E. L. Pellini. A model-based and goal-oriented approach for the conceptual design of smart grid services. *Machines*, 2021. https://doi.org/10.3390/machines9120370.

[35] I. Osman, S. B. Yahia, and G. Diallo. Ontology integration: Approaches and challenging issues. *Information Fusion*, 2021. https://doi.org/10.1016/j.inffus.2021.01.007.

[36] R. Pergl, T. P. Sales, and Z. Rybola. Towards OntoUML for software engineering: From domain ontology to implementation model. In *Model and Data Engineering (MEDI)*, 2013. https://doi.org/10.1007/978-3-642-41366-7_21.

[37] Y. Purnomo, R. Doss, N. B. Suhardi, and N. B. Kurniawan. Consolidating service engineering ontologies building service ontology from SOA modeling language (SoaML). *International Journal of Computer and Information Engineering*, 2018. https://doi.org/10.1109/ICITSI.2018.8695936.

[38] G. Reggio. A UML-based proposal for IoT system requirements specification. In *International Workshop on Modelling in Software Engineering (MiSE)*, 2018. https://doi.org/10.1145/3193954.3193956.

[39] C. Reginato, J. Salamon, and M. P. Barcellos. Ontology integration approaches: A systematic mapping. In *CEUR Worshops*. CEUR-WS.org, 2018.

[40] Z. Rybola and R. Pergl. Towards ontouml for software engineering: Transformation of rigid sortal types into relational databases. In *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2016. https://dx.doi.org/10.2298/CSIS170109035R.

[41] F. M. Suchanek. OntoUML specification. https://ontouml.readthedocs.io/en/latest/, 2018.

[42] M. Tabatabaie, F. A. C. Polack, and R. F. Paige. KAOS-B A goal-oriented process model for EIS. In *International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (ICEIS)*, 2010. http://dx.doi.org/10.5220/0003016000400049.

[43] S. Tueno, R. Laleau, A. Mammar, and M. Frappier. Towards using ontologies for domain modeling within the SysML/KAOS approach. In *International Requirements Engineering Conference Workshops (REW)*, 2017. http://dx.doi.org/10.1109/REW.2017.22.

[44] A. van Lamsweerde. The KAOS meta-model: Ten years after. Technical report, Universite Catholique de Louvain, 1993.

[45] V. Werneck, A. de Padua Oliveira, and J. C. S. do Prado Leite. Comparing GORE frameworks: i-star and KAOS. In *Workshop on Requirement Engineering (WER)*, 2009.

[46] F. Zickert. Evaluation of the goal-oriented requirements engineering method kaos. In *Americas Conference on |Information Systems (AMCIS)*, 2010.

Predicting Covid-19 Infections With a Multi-Agent Organizational Approach and Machine Learning Techniques

Samir Safir and Abderrahim Siam
ICOSI Laboratory, Computer Science Department, Abbes Laghrour University, Khenchela BP 1252 El Houria, Algeria
E-mail: safir.samir@univ-khenchela.dz, siamabderrahim@gmail.com

Student paper

Keywords: Multi-agent system, machine learning, predictive analytics, healthcare, machine learning

Received: April 1, 2023

Our study presents a strategy for designing and implementing a Multi-Agent System (MAS) using organizational paradigms. The developed system offers a healthcare-oriented approach that utilizes the Internet of Medical Things (IoMT) to assist public health authorities in predicting COVID-19-infected patients. The proposed approach leverages autonomous agents to handle dynamic data from various sources within a structured organization. These agents collaborate to make effective, real-time predictions. As the agents continuously learn from the cases entering the system, the accuracy of predictions improves over time. The system was implemented using the JaCaMo framework, which integrates three key layers of MAS programming: organization, environment, and agent programming. The methodology demonstrated a prediction accuracy of over 90%, outperforming state-of-the-art (SOTA) approaches by enabling faster real-time decision-making. This capability facilitates the efficient processing of real-time big data, making a significant contribution to the advancement of predictive healthcare systems.

Povzetek: Razvit je sistem za napovedovanje okužb s COVID-19 z uporabo večagentnega sistema (MAS) in algoritmov strojnega učenja. Sistem omogoča natančne napovedi z analizo podatkov v realnem času, izboljšanih z učnimi agenti in IoMT.

1 Introduction

Artificial intelligence [1] is becoming increasingly important in healthcare and has the potential to revolutionize the way we diagnose, treat, and prevent diseases. In this paper, our aim is to develop an intelligent system to predict COVID-19 infection cases by involving a multitude of concepts, such as Multi-Agent Organization, IoT devices, Machine Learning algorithms, and Big Data analytics. This paper presents a novel approach that leverages these advancements to address the urgent need for accurate COVID-19 infection predictions.

Agent-Oriented Engineering is a widely recognized method for constructing distributed and complex software systems. This approach focuses on using autonomous, proactive agents as the key elements in the design and development process, making it well-suited for systems that operate in highly dynamic environments. Autonomy is a fundamental characteristic of agents in Multi-Agent systems (MAS). However, while the autonomy of agents can be beneficial in many contexts, it may also lead to challenges, such as dispersed behavior that prevents alignment with global objectives. Consequently, conventional Multi-Agent models frequently adopt an individualistic outlook towards the environment by treating agents as self-governing entities pursuing their objectives based on their perceptions and abilities. In critical applications, such as those found in business or government settings, it is essential to consider the behavior of the overall system.

The aim of Multi-Agent systems research is to understand how autonomous agents can collaborate to solve problems and create collective outcomes that cannot be achieved by each agent working alone [2]. To reduce the dispersing effects of agent autonomy, organizations offer solutions. A Multi-Agent Organization is a social entity composed of multiple agents, structured according to specific topologies and communication relationships. These agents work together to complete multiple tasks to fulfill the overall goal of the organization [3].

To attain global objectives in MAS, the independent behavior of individual agents may need to be regulated. This is where organizational models come into play, serving as a means of controlling agent behavior so that they can collaborate effectively to attain shared objectives [4]. These models can be divided into two perspectives [5]: Agent-Centered MAS, where the designer focuses on the behavior of individual agents and their interactions without considering the system's structure, and Organization-Centered MAS, which emphasizes the behavior of the system as a whole. From this perspective, the designer considers both the overall organizational structure and coordination patterns, as well as the individual actions of each agent.

By creating norms or guidelines, organizational abstraction facilitates the coordination of agents' local behavior and interactions with one another. Furthermore, this model enables agents to reason about the overall organizational structure and the behavior of other agents.

Several applications based on Multi-Agent systems that utilize Organization Theory have demonstrated its usefulness and proven successful in various fields, including simulation, e-commerce, network management, collective robotics, avionic mechanical design, traffic simulation, and more. These diverse applications underscore the versatility and effectiveness of multi-agent systems across various domains. Additionally, the organization in a Multi-Agent system is important to support adaptation to environmental changes. These changes may be addressed by transitioning from one organization to another (reorganization or self-organization), as evidenced by the significant amount of research in this area.

The Internet of Things (IoT) is a network of physical objects, devices, and sensors that are connected to the internet and can communicate and exchange data with other devices or systems. IoT technology enables the collection and analysis of large amounts of data, which can be used to improve operational efficiency, reduce costs, and enhance decision-making. More specifically, we are interested in the Internet of Medical Things (IoMT) [6], which refers to medical devices, sensors, and wearables that are connected to the internet and can exchange health-related data. This interconnectedness plays a pivotal role in enhancing healthcare delivery.

In the proposed approach, we combine the concepts and techniques presented above with Big Data Analytics and machine learning algorithms [7]. Big Data Analytics refers to the process of analyzing large and complex datasets to extract insights, patterns, and trends that can help inform decision-making. It involves using advanced tools and techniques to process and analyze data from various sources [1], including structured and unstructured data, to uncover meaningful insights.

With these foundational concepts established, the subsequent sections of this paper will detail the proposed approach and its implementation. Section 2 presents a motivating healthcare example, while Section 3 reviews related research across various fields. Section 4 compares our approach with existing methods from related work, and Section 5 focuses on the design of the proposed solution. Section 6 addresses the implementation and utilization of the developed approach, and Section 7 provides a comparison and discussion of results from each agent in the system. Finally, the "Conclusions and Future Work" section offers concluding remarks and potential directions for future research.

2 Motivation and overview

To demonstrate the importance of incorporating organizational perspectives into the design of Multi-Agent Systems (MAS), we present a scenario envisioning a solution for combating the Coronavirus pandemic. This study introduces an architecture for a Multi-Agent System that leverages machine learning algorithms to rapidly identify Covid-19-infected patients. Our approach is based on a real-time investigation system that collects physiological data from patients, including body temperature, ECG, heart rate, oxygen levels, blood pressure, glucose levels, and more.

The system relies on accurate information from hospitals connected to it, necessitating an efficient method for storing and processing large volumes of data. Big Data technology is utilized to digitally store comprehensive information on all Covid-19 cases, including those currently infected, recovered, or deceased. The stored data can be continuously analyzed to develop future preventive measures. The system applies a Multi-Agent Organizational model, enhanced with data analytics powered by machine learning (ML), to analyze the collected data and improve prediction models in real-time.

This solution aids local health authorities in monitoring a large number of users, promptly alerting them if symptoms are reported. Health officials can then reach out to the affected users, instructing them to report to the hospital for testing. Patients are admitted for observation until test results are confirmed. Additionally, the system tracks individuals in close contact with the infected patient, including family members, friends, and coworkers, and monitors them for any signs of infection.

By assisting health authorities in controlling the spread of Covid-19, this approach helps alleviate the burden on medical staff. Furthermore, the solution is adaptable for use in other hazardous pandemics or public health crises, offering a versatile and scalable tool for managing health emergencies.

3 Related work

As noted by Ilana et al. (2021) [8], the majority of AI research aimed at combating the coronavirus can be classified into four main categories: diagnosis and prognosis, treatments and vaccines, social control and tracking, and prediction. This paper focuses primarily on the latter category—prediction.

Otoom et al. (2020) [9] proposed a system for detecting and monitoring Covid-19 cases in real-time. During quarantine, IoT devices were deployed to gather real-time physiological data, and machine learning algorithms were used to enhance predictive accuracy. Their study compared seven machine learning algorithms, with five showing improvements in prediction accuracy. This demonstrates the effectiveness of using IoT in combination with machine learning for real-time pandemic response.

A survey by Thanh (2020) [10] analyzed various AI techniques applied to combat Covid-19, focusing on data analytics, natural language processing, and data mining. The work highlighted the diversity of AI applications in pandemic management, particularly in addressing key challenges such as big data processing and decision-making efficiency.

Carrillo et al.(2020)[11] employed unsupervised machine learning techniques, including k-means clustering, to classify countries based on similar Covid-19 infection patterns. This study emphasized the importance of using machine learning to understand the geographic spread of the virus and revealed significant insights into transmission trends across different regions.

Janko et al.(2021)[7] explored how non-countermeasure factors, such as culture, development, and travel, contributed to the early spread of Covid-19 before strict interventions were implemented. Using machine learning and statistical models, the study achieved approximately 80% prediction accuracy, showing how interconnected societal factors affect viral transmission in diverse contexts.

Recently, agent-based systems have emerged as a promising approach to addressing limitations in the healthcare sector. By integrating Multi-Agent Systems (MAS) into medical applications, healthcare costs can be reduced, and the burden on medical professionals can be alleviated. These systems shift the focus toward preventive, long-term care, which is patient-centered rather than hospital-centered. This transformation includes the use of remote monitoring systems that enable patients to play a more active role in managing their health and treatment, especially during extended care periods in both hospitals and homes.

Isern et al.(2016)[12] developed a MAS-based platform for managing patient care during hospital stays. This platform collects real-time data from various sources, facilitating dynamic bed occupancy allocation, doctor assignments, medical procedure planning, and automated billing by tracking the behavior of hospital actors in real time. The system demonstrates the potential of MAS to optimize healthcare operations and improve hospital management efficiency.

Lanzola et al.(1999)[13] proposed a framework for developing interoperable Multi-Agent Systems for medical applications. Their work highlighted the need for MAS in enhancing collaboration across diverse healthcare systems. Additionally, Juan et al.(2006) [14] in an Ambient Intelligence (AmI) ecosystem for Alzheimer patients, another study by Gonzalez et al.(2002)[15] developed BDI agents that integrate context-aware technologies to gather real-time data from users, further advancing patient-centered care.

4 Discussion

To summarize the key methodologies, results, and contributions from the aforementioned studies, we present a comparative table (Table .1). This table provides a clear overview of the different approaches to Covid-19 prediction, their key features, and their relative performances.

This table highlights how different methods approach COVID-19 prediction, focusing on collaboration efficiency, machine learning techniques, and accuracy. It also demonstrates the strength of our proposed MAS-based system, particularly its adaptability, collaboration-driven improvements, and real-time monitoring capabilities, which are critical for managing future pandemics and healthcare crises.

In summary, previous research demonstrates the effectiveness of AI and MAS in addressing various healthcare challenges, particularly in pandemic response and patient monitoring. Our study builds on these approaches by integrating Multi-Agent Organizational paradigms with IoMT devices and machine learning for real-time Covid-19 prediction, providing a novel solution that enhances predictive accuracy and supports public health efforts.

5 The proposed solution

Several studies conducted in the field of Multi-Agent learning have emphasized the importance of large datasets [16, 17]. Additionally, extensive research has been performed on various models, such as ensembles of classifiers [18, 19]. The goal of this research is to develop a predictive Covid-19 case detection system using a Multi-Agent approach based on an organizational model. The system operates in a distributed environment and is composed of different sites referred to as ”Assistant Controllers” (ACs). Each AC consists of two agents: a Learner Agent and an Interface Agent. It is crucial to note that different sites may have distinct instances of datasets. The knowledge generated by the independent ACs at each site will be consolidated into a single knowledge repository. The global system comprises of a collection of ACs and a special agent called a Broker agent or Mediator, as illustrated in (Fig. 1).

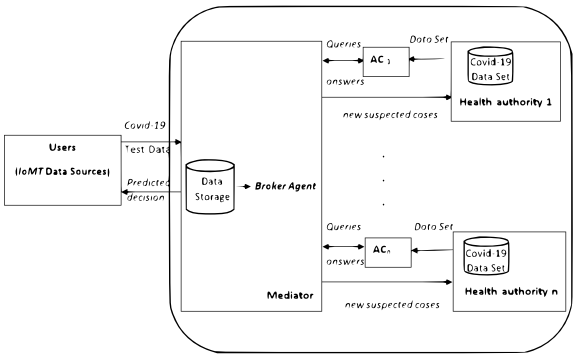


Figure 1: The global architecture of MAS organization Covid-19 approach

The agents communicate and share information about the selected datasets at each node. Subsequently, the Broker Agent selects the dataset that provides the most significant

Table 1: Performance comparison of predictive methods

Article Title	Key Metrics/Variables	Methods	Results/Findings
[9]:An IoT-based Framework for Early Identification and Monitoring of Covid-19 Cases	Accuracy, real-time health data	IoT, Ensemble Learning (RF GBTs), Apache Spark	Achieved over 90% accuracy with various AI algorithms; proposed a real-time framework for monitoring Covid-19 using IoMT and WBANs.
[11]:Using Country-Level Variables to Classify Countries According to Confirmed Covid-19 Cases	Covid-19 case counts, country variables	Unsupervised Machine Learning	Identified distinct clusters of countries based on Covid-19 case counts using unsupervised learning techniques.
[13]:A Framework for Building Cooperative Software Agents in Medical Applications	Collaboration efficiency, application areas	Framework Development, Agent-Based Modeling	Proposed a cooperative agent framework for healthcare applications to enhance collaboration among medical agents.
[14]:Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care.	Patient monitoring metrics, AI integration	Agent Technology, Sensor Networks	Developed an intelligent monitoring environment for Alzheimer's patients to enhance care and support for families.
[7]:Machine Learning for Analyzing Non-Countermeasure Factors Affecting Early Spread of Covid-19	Spread factors, machine learning accuracy	Machine Learning, Data Analysis	Analyzed non-countermeasure factors affecting Covid-19 to spread; developed predictive models with promising accuracy 80% in identifying influential factors.
The proposed approche	Collaboration efficiency ,Decision-making efficiency , machine learning accuracy , real-time health data	Multi Agent based on an Organizational Modeling, Incremental Machine & ensemble learning Techniques	Support for Remote Care, Real-time Monitoring and Alerts, Improved Decision-Making, Customizable for Future Pandemics

information. After this, the prediction decisions are sent to health authorities and users for preventive action (see Fig. 2). In our proposed approach, the agents work together within a structured organization to coordinate their behavior and cooperate to achieve the global objective of improving Covid-19 predictions.

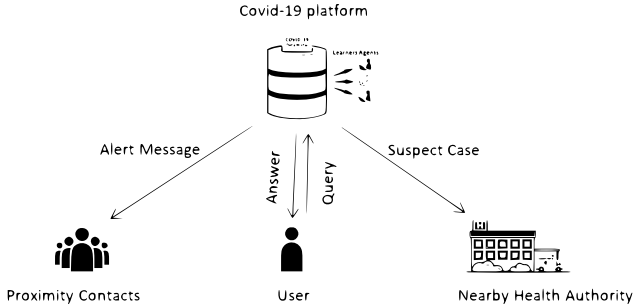


Figure 2: The global architecture of the MAS organizational approach to Covid-19 prediction

5.1 Assistant controller AC

An Assistant Controller (AC) is a Multi-Agent System (MAS) designed to categorize and predict Covid-19 patients using streaming datasets provided by health authorities. The structure of the AC is illustrated in Fig. 3. The Assistant Controller (AC) operates within the framework of the Agent and Artifact (A&A) paradigm, as discussed in Section 5.2.3. This paradigm serves as a conceptual model that outlines the interactions between autonomous agents and their environment, facilitating effective communication and collaboration.

At the heart of the AC's operation are two primary agents:

1. *Interface Agent:* This agent is responsible for capturing every incoming dataset at the Assistant Controller (AC) and processing bid queries from the Broker Agent, ensuring that relevant data is available for decision-making. Upon receiving new data or queries, the Interface Agent immediately notifies the Learner Agent by updating its environment through the Agent

and Artifact (A&A) mechanism, where artifacts act as intermediaries to facilitate communication, coordination, and collaboration among agents. This process initiates the prediction process, ensuring continuous and adaptive updates to the predictive model.

2. *Learner Agent:* This agent plays a crucial role in updating the predictive model. Upon receiving notifications from the Interface Agent, the Learner Agent incrementally updates the model based on the incoming data. It can also predict the queried bids as the expected results. This capability allows the Learner Agent to refine its predictions continuously, ensuring that the model remains current and relevant.

The real-time coordination between the Interface Agent and the Learner Agent enables the AC to continuously rebuild and dynamically update the predictive model. As a result, our system effectively manages highly dynamic healthcare datasets, adapting seamlessly to new information as it becomes available. This flexibility is essential in a rapidly changing context like Covid-19, where timely and accurate predictions can significantly impact public health responses.

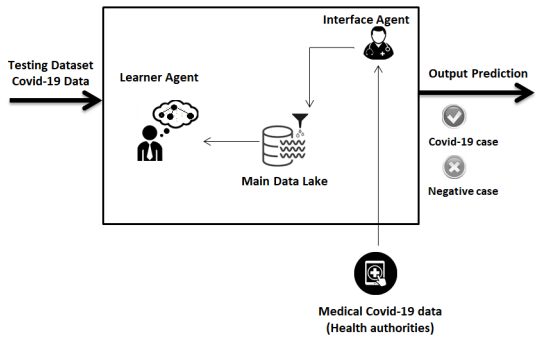


Figure 3: Assistant Controller architecture

5.1.1 Interface agent

The Interface Agent plays a crucial role in cleaning, standardizing, and transforming data to produce high-quality datasets. Preparing these training datasets is essential for machine learning and statistical analysis, as it allows a computer to acquire knowledge and effectively learn problem-solving techniques. Both machine learning models and statistical analyses rely heavily on these well-prepared datasets.

Since datasets may contain erroneous or missing information, data cleaning is necessary to remove or correct these issues. The data is partitioned into separate sets for testing and training purposes, with the machine learning (ML) model being trained on 75% of the dataset. The efficiency of the proposed solution is then evaluated using the remaining 25%. Additionally, the Interface Agent serves as a connection point between the Learner Agent and the Broker Agent, facilitating seamless communication and coordination within the system.

Medical data streams: Whenever a new Covid-19 case is discovered, it must be reported promptly to the relevant Assistant Controller system. This allows the intelligent learning agents to update the model using only a single iteration through the data, leading to better and more accurate real-time predictions.

5.1.2 Learner agent

The Learner Agent has two vital roles to fulfill. The first function is responsible for creating the training model, referred to as the Classify function. The second function is tasked with making predictions, known as the Predict function. This function utilizes the model to predict unknown outcomes. Notably, the goals of these two missions can be achieved in parallel, leveraging the strengths of a Multi-Agent System (MAS) and incremental machine learning techniques.

Supervised machine learning algorithms include a classification type, where a model is developed from a set of labeled training data. As a result, the model can predict the class or label for newly discovered data, known as testing data. Metrics such as accuracy, which are calculated based on the testing results, can be used to evaluate classification performance . Numerous classification algorithms exist, including decision trees, random forests, neural networks, rule-based algorithms (such as conjunctive rules and PRISM), logistic regression, naive Bayes, and others. Each algorithm has its distinct advantages and limitations, and the selection of an appropriate algorithm depends on various factors, including the data's attributes and the desired objectives [20].

Decision trees are highly regarded classification techniques in the field of data mining and have applications across various domains, including business intelligence, biomedicine, and healthcare. The conventional method of creating a decision tree is known as the Greedy Search approach. This process involves loading the complete dataset into memory and organizing it into a series of nodes and leaves that form a hierarchical structure. However, a significant drawback of this method is that once the decision tree has been constructed, it cannot be easily modified or updated, even in the presence of new data. Incorporating newly acquired information necessitates the complete reconstruction of the tree by loading both historical and recent data, a process that may be time-consuming and computationally intensive. Additionally, this approach risks losing critical information or patterns that were present in the original tree. To address these limitations, various alternative methods for decision-tree learning have been developed to enhance the scalability and adaptability of decision trees to changing data [21].

The conventional approach is inadequate for handling limitless data inputs, such as data streams, where information arrives in real-time. To tackle this challenge, an incremental approach has been introduced, enabling dynamic model construction so that the tree expands as new data is

inputted. The Very Fast Decision-Tree (VFDT) algorithm, proposed by [22], employs the Hoeffding Bound [23] for incremental node splitting. This approach constructs a decision tree by continuously observing the features of incoming data and maintaining a record of their statistics. Once sufficient statistics are collected from every leaf, a node-splitting algorithm is applied to determine whether there is adequate statistical evidence to justify a node split. If a node split is warranted, the tree is expanded, and a new decision node is added in place of the leaf. This technique allows the decision tree to learn continuously as the data stream is processed in real-time [24]. The primary advantage of this approach is its ability for real-time data mining, eliminating the need to store all data in advance since data streams are potentially unlimited. Unlike traditional decision trees that require retraining with every new data arrival, the incremental approach allows for dynamic updates to the tree, enabling more efficient and effective data analysis. The process of building a VFDT is demonstrated in Algorithm 1 [25]. Notably, node divisions across multiple leaves can occur concurrently and autonomously [26].

While several libraries in the literature support incremental learning—particularly for the Very Fast Decision Tree (VFDT), such as MOA, Apache Spark MLlib, Scikit-Multiflow, and River—we have chosen WEKA for the following reasons:

1. *Compatibility with Java Eclipse:* WEKA is fully compatible with Java, allowing for seamless integration with the Java Eclipse IDE. Its rich package and API facilitate the easy incorporation of WEKA's machine learning capabilities into larger Java applications, making it a versatile tool for implementing machine learning within the JaCaMo framework.
2. *User-Friendly Interface:* WEKA provides an intuitive interface for data preprocessing, model evaluation, and visualization. This streamlines the machine learning workflow, making it accessible for researchers who are new to the field.
3. *Community Support and Documentation:* WEKA boasts a strong community and extensive documentation, which facilitate troubleshooting and ensure the availability of resources for further experimentation.
4. *Suitability for Multi-Agent Systems:* WEKA's capabilities align well with the requirements of our multi-agent system, particularly in handling large datasets, performing real-time predictions, and integrating seamlessly with the JaCaMo framework.
5. *Runtime Metrics:* We will present detailed runtime metrics for WEKA's machine learning tools to demonstrate their efficiency, including average training and testing times for various algorithms implemented within WEKA, such as Random Forest, VFDT, and others utilized in our study.

We selected the Very Fast Decision Tree (VFDT) algorithm for several compelling reasons relevant to healthcare applications. First, VFDT is designed for high scalability [27], enabling it to efficiently manage large-scale data streams typical in healthcare settings. Unlike ensemble-based approaches, which often require full retraining, VFDT allows for incremental updates, making it well-suited for the rapid influx of patient data. Additionally, VFDT's real-time adaptation capabilities [28] ensure that the model can quickly adjust to dynamic changes in healthcare datasets. Finally, its memory efficiency [29] enables effective resource management, which is crucial in healthcare environments where computational power may be limited. Thus, the combination of efficiency, adaptability, and memory management makes VFDT a more suitable choice for our predictive modeling needs compared to ensemble-based incremental algorithms.

Algorithm 1: Very Fast Decision Tree Induction

Input: S: A sequence of examples; X: The set of attributes; γ: One minus the desired probability; ε: The Hoeffding bound
Output: τ : very fast decision tree learned from S

```
Initialize τ with a single root node.
Initialize the statistics for tree growth.
foreach s ∈ S do
    Sort s into leaf node l using τ.
    Update the statistics at l for tree growth.
    if Examples at l are not from the same class then
        foreach attribute X_a ∈ X do
            Calculate the Hoeffding bound using the formula: ε = sqrt(R^2 ln(1/δ) / 2|S|) (where R is the range of the attribute values);
            Select the attribute with the highest Hoeffding bound and split l on that attribute.
            if multiple attributes have the same highest Hoeffding bound then
                select the one with the highest information gain.
```

5.2 Broker agent

The system continuously receives incoming queries related to Covid-19 testing datasets for prediction. To facilitate communication between the collection of Assistant Controller (AC) agents and external entities, an intermediary agent known as the Broker Agent is employed. The Broker Agent is responsible for mediating interactions between the AC agents and the external environment.

Once it receives the queries, the Broker Agent utilizes Random Forest techniques to select the most appropriate predictions based on the aggregated data from the AC agents. This approach ensures that the predictions are robust and take advantage of the diverse insights provided by the various AC agents.

5.2.1 Random forest

The Random Forest algorithm is an ensemble learning method that involves building multiple decision trees and combining their predictions to enhance accuracy. Each decision tree is constructed using a randomly selected subset of the training data and a random subset of features. The final prediction is determined by averaging the predictions from all the trees in the forest. This approach helps reduce overfitting and improves the overall accuracy of the model. Based on the principles of Random Forest classification [30], the proposed approach involves selecting the most relevant majority vote decision from various Assistant Controller (AC) groups. This method operates under the assumption that different AC groups may have access to different sources of information and may employ distinct decision-making criteria. By combining the decisions from multiple AC groups, we can leverage their diverse perspectives and expertise to arrive at a more informed and accurate conclusion.

We chose Random Forest as the primary decision-making technique at the broker agent level due to its strong performance in managing complex, high-dimensional datasets typical of healthcare applications. Although Random Forest presents challenges in interpretability—especially when dealing with distinct datasets from each AC—its robustness and ability to handle missing data (particularly when one AC does not respond), along with its capacity to provide actionable insights, make it a strong candidate for our predictive modeling efforts in the healthcare domain.

To implement this approach, we first need to identify the relevant AC groups that can contribute input to the decision-making process. Various criteria can be employed to select these groups, including their area of expertise, past performance, or level of authority.

Once the relevant AC groups are identified, we can gather their decisions and use a voting mechanism to determine the most pertinent majority decision. This can involve assigning weights to the decisions based on the expertise or performance of each group or utilizing a more complex algorithm to combine decisions in a meaningful manner.

Overall, this approach has the potential to improve the accuracy and robustness of the decision-making process by leveraging the collective intelligence of multiple AC groups. However, it also requires careful coordination and communication between the different groups to ensure that their decisions are aligned and consistent with the overall goals and objectives of the organization.

5.2.2 Contract Net

To prevent potential bottlenecks associated with the Broker Agent, the system employs the Contract Net Protocol, originally proposed by [31] and later refined by [32]. This protocol is designed to facilitate the announcement of transferable tasks and to solicit bids from Interface Agents representing each Assistant Controller (AC) group capable of executing these tasks.

When a task is announced, the Interface Agents respond with bids that indicate their perceived capability to fulfill the task. The Broker Agent collects these bids and allocates the task to the bidder with the highest offer. Although the Contract Net Protocol is often viewed as a negotiation technique, its primary function is to act as a coordination mechanism for task allocation.

Task Announcement and Bid Solicitation: When a specific task arises—such as processing a new Covid-19 dataset for prediction—the Broker Agent announces this task to the relevant AC groups within the system. Each Interface Agent receives the announcement and evaluates its own capabilities, resources, and current workload before deciding whether to respond with a bid. The bid submitted by each Interface Agent includes not only an estimate of the time and resources required to complete the task but also their confidence level in executing it successfully.

Bid Evaluation and Task Allocation: Once the Broker Agent has received all the bids from the Interface Agents, it conducts a thorough evaluation to determine which bid offers the best combination of capability and cost. The Broker Agent allocates the task to the bidder with the highest offer, taking into account the quality of the bid and the capabilities of the Interface Agent. This allocation process ensures that tasks are assigned to agents that are most likely to perform them effectively and efficiently.

Coordination and Flexibility: Although the Contract Net Protocol is often viewed primarily as a negotiation technique, its primary function is to serve as a coordination mechanism for task allocation. By allowing dynamic task allocation and enabling agents to bid for multiple tasks simultaneously, the protocol enhances the system's flexibility and adaptability. This is particularly important in a healthcare setting where the demands on the system can vary significantly and unpredictably.

Workload Balancing The protocol also promotes workload balancing within the system. Since agents that are already busy with other tasks are less likely to place bids, the distribution of tasks tends to be more even across the available agents. This self-regulating mechanism helps to prevent any single agent from becoming overwhelmed with too many tasks, thereby maintaining overall system performance.

Limitations of the Contract Net Protocol: However, the Contract Net Protocol does have its limitations:

1. *Conflict Detection and Resolution:* One notable drawback is that the protocol lacks built-in mechanisms for conflict detection and resolution. If two agents bid for the same task or if their bids conflict in terms of resource availability, the protocol does not provide a framework for resolving these conflicts, which may lead to inefficiencies or delays.
2. *Heavy Reliance on Communication:* The protocol heavily relies on communication between agents. This reliance can introduce delays or inefficiencies in task allocation, especially in scenarios where network la-

tency or communication failures occur. If the communication channels are congested or disrupted, it could hinder the timely allocation of tasks.

3. *Scalability Concerns:* In larger systems with numerous agents and tasks, the process of soliciting bids and evaluating them can become computationally expensive and time-consuming. This could potentially lead to performance bottlenecks as the system scales.
4. *Limited Bidder Pool:* The effectiveness of the protocol is contingent upon having a sufficient number of agents willing and able to submit bids. If too few agents respond to a task announcement, it could result in suboptimal task allocation.

Overall, while the Contract Net Protocol offers a structured and efficient means of task allocation within the Broker Agent framework, it is important to be aware of these limitations and to consider supplementary mechanisms or strategies that could mitigate potential issues and enhance the overall robustness of the system.

5.2.3 Agents and artifacts (A&A) environment

Our proposal involves utilizing the Agents and Artifacts meta-model to create a coordination mechanism for developing a shared environment, as described by [33]. This model addresses some of the shortcomings of the Contract Net Protocol by representing the environment as workspaces, where Agents and Artifacts coexist to provide various services. In this environment, agents interact with artifacts through a mechanism known as the focus action. When an agent focuses on an artifact, it gains access to the observable properties of that artifact as perceptions. Additionally, artifacts can offer a range of operations or actions that agents can perform, enhancing collaboration and interaction within the system. *Bidding Process for Task Allocation:* The Broker Agent plays a crucial role in locating Interface Agent contractors for each Assistant Controller (AC) group (Fig. 4). This is achieved through a structured bidding process, which follows these steps:

1. The Broker Agent announces the task (the task is considered an artifact in the system).
2. Interface Agents evaluate the task based on their capabilities and commitments.
3. Interface Agents submit bids to the Broker Agent.
4. The Broker Agent evaluates the received bids and selects the majority vote from the competing predictions.

Decentralized and Collaborative Approach: This bidding scheme is completely distributed, meaning that each agent operates independently without a centralized controller. The Broker Agent facilitates the announcement of tasks and the collection of bids, but the decision-making process relies on the collective actions of multiple agents. Each Interface Agent autonomously assesses the task based on its own capabilities and commitments, leading to a diverse range of bids that reflect varying perspectives. The final decision regarding task allocation is made based on the majority of predictions from the Interface Agents. This approach not only ensures a decentralized and collaborative handling of tasks but also leverages the diverse expertise of multiple agents to enhance the accuracy and reliability of the outcomes.

By employing the Agents and Artifacts framework, we create a flexible and adaptive environment that can effectively manage the dynamic and complex nature of healthcare data processing tasks. The interaction between agents and artifacts allows for improved coordination, resource utilization, and task execution in the context of predicting Covid-19 patient outcomes.

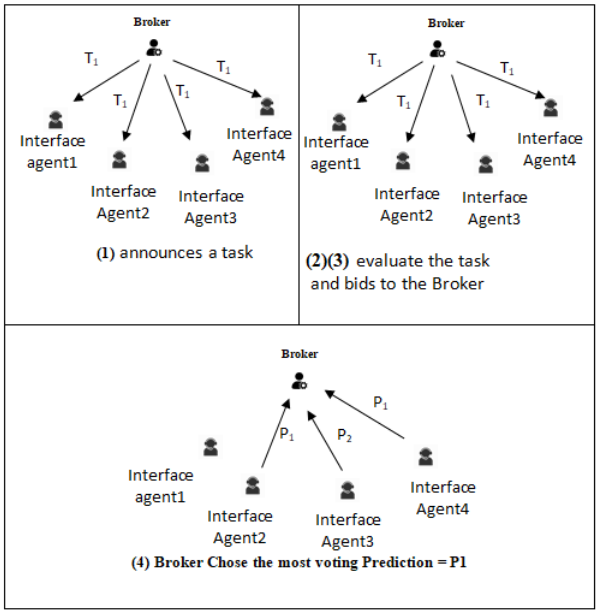


Figure 4: Broker agent

6 Some implementation aspects

The integration process described in the previous sections was implemented using the Eclipse Modeling and JaCaMo framework. The JaCaMo[1] framework is a multi-agent-oriented programming platform (MAOP) designed to facilitate the development of complex multi-agent systems [34]. This framework utilizes autonomous agents programmed in Jason[2] , which operate within environments defined as artifacts in CArtAgO [3]. Additionally, these agents are organized by the Moise framework [4], and they interact with each other using a specified interaction language.

All experiments were conducted on a system equipped with an Intel Core i5-1035 G1 processor running at 1.00 GHz and 16 GB of RAM. The software application Alert-Covid employs the proposed model, consisting of three Assistant Controllers named UK, Biskra, and ESI. The Broker

[1] http://jacamo.sourceforge.net
[2] http://jason.sourceforge.net/wp
[3] http://cartago.sourceforge.net
[4] http://moise.sourceforge.net/

Agent serves as the intermediary, facilitating communication and coordination among the different Assistant Controllers.

6.1 UK assistant controller

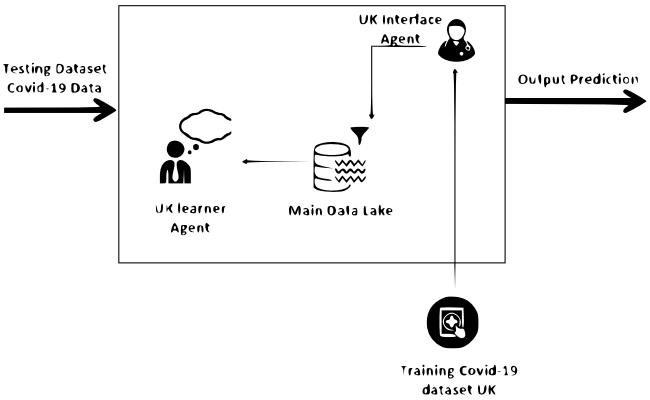


Figure 5: UK architecture

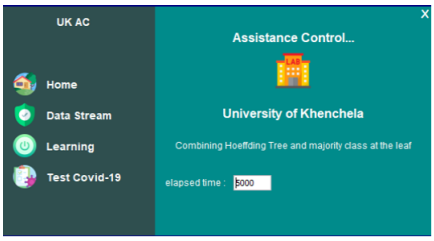


Figure 6: UK interface

6.1.1 UK interface agent

For the training data at the UK Assistant Controller site (Fig 7), we utilize the dataset titled Covid-19 Symptoms and Presence, which is publicly available on Kaggle [35]. This dataset comprises 20 attributes that represent potential factors associated with contracting Covid-19, including symptoms such as cough and fever, along with other relevant indicators. Additionally, it contains a class attribute indicating the presence of the virus.

The dataset is sourced from reputable organizations, including the World Health Organization and the All India Institute of Medical Sciences, with the aim of facilitating research into the prevalence of symptoms and their correlation with Covid-19 diagnoses. Its comprehensive nature makes it a valuable resource for predictive modeling in the healthcare domain.

6.1.2 UK learner agent

The UK Assistant Controller's learning agent implement the Hoeffding tree algorithm, which is integrated into the open-source software WEKA, to obtain knowledge from

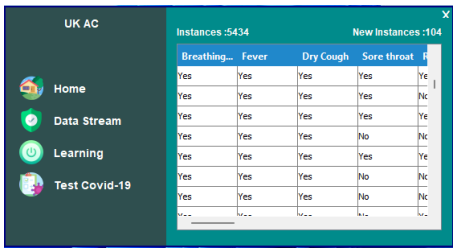


Figure 7: UK interface agent

data streams. WEKA is a suite of machine learning algorithms developed at the University of Waikato in New Zealand [36]. It facilitates data mining by providing a wide range of tools for clustering, pre-processing, visualization, association, regression, and classification.

The VFDT implementation in the UK learner agent uses the following specific parameters Fig. 8:

1. *Splitting Criterion:* Gini Index, which is a measure of the impurity or variance of a node. The algorithm selects the attribute with the lowest Gini Index to split the node, leading to purer child nodes.
2. *Allowable Error* δ*:* $1 \cdot 10^{-7}$.This parameter controls the confidence level for deciding whether to split a node. A lower error threshold ensures higher confidence in the decision to split, minimizing the likelihood of unnecessary splits, which helps in maintaining the efficiency of the model.
3. *Grace Period:* 100 instances. This setting ensures that the algorithm waits for 100 instances before evaluating a potential split at any node. This helps to accumulate sufficient data to make statistically sound splitting decisions, avoiding premature splits based on insufficient data.

These parameters help the UK learner agent achieve a balance between accuracy and efficiency, making it capable of handling continuous data streams while ensuring the decision tree adapts quickly to new data without overfitting.

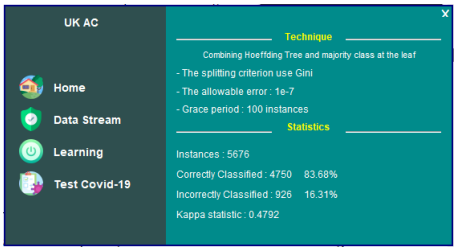


Figure 8: UK learner agent

A learning curve plots accuracy against the number of instances, as illustrated in (Fig. 9), showing how the model's performance improves with more training data.

- The accuracy starts at 81.42% at 5602 instances and fluctuates slightly as more data is processed.
- Over time, there is a general upward trend, with accuracy reaching 93.65% at 1982 instances.

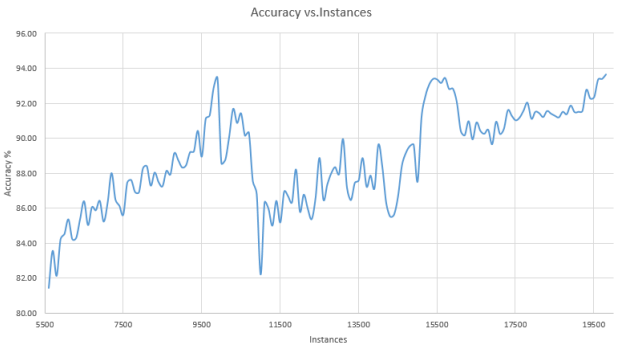


Figure 9: UK learner curve

– The model demonstrates significant improvement, achieving over 90% accuracy as it processes larger datasets.

Analysis:

– The learning curve likely depicts an overall positive trend with some small fluctuations. These fluctuations suggest that the model is learning from new data but may occasionally encounter challenges from certain instances.
– Despite these variations, the general increase in accuracy indicates the model's capacity to learn and improve its predictions as more instances are introduced.

The UK learner model shows a general improvement in accuracy as more data is processed, starting at 81.42% and reaching over 90%. Despite minor fluctuations, the overall trend is upward, indicating the model's ability to adapt and learn effectively from increasing data.

6.1.3 UK test Covid-19 case

With the implementation of our proposed architecture, we developed a GUI module for testing the UK Assistant Controller, enabling functional testing at the local level. Fig.10 presents an example of a positive Covid-19 case.

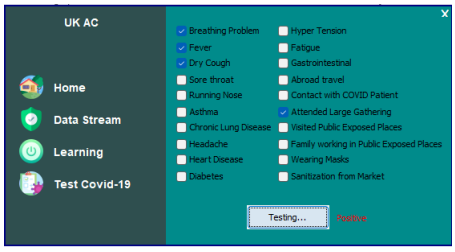


Figure 10: UK test Covid-19 case

6.2 ESI assistant controller

6.2.1 ESI interface agent

The ESI Interface Agent operates as a crucial component within the multi-agent system, specifically tasked with handling and processing COVID-19-related health data from a localized data stream. To enable effective COVID-19 prediction, the ESI Interface Agent utilizes a dataset sourced from a research repository [37]. This combined dataset, titled "Covid Symptoms," features 3,021,444 entries and ten essential characteristics related to COVID-19 symptoms and test outcomes. The dataset includes key information on symptoms such as cough, fever, shortness of breath, sore throat, and headache, which are critical indicators for COVID-19 detection.

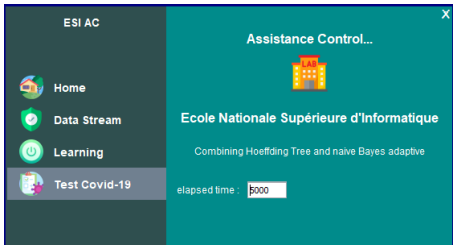


Figure 11: ESI interface

Given the dataset's comprehensive nature, the ESI Interface Agent processes and cleans the data to ensure its usability for predictive modeling. The cleaned and preprocessed data is then forwarded to the Learner Agent, which uses the Very Fast Decision Tree (VFDT) algorithm for incremental learning. This approach allows the agent to continuously update the prediction model in real time as new COVID-19 cases are added to the stream, without the need for retraining from scratch.

The choice of this dataset aligns perfectly with the goals of the ESI Interface Agent for several reasons:

1. *Rich Symptom Features:* The dataset contains detailed records of COVID-19 symptoms, allowing the ESI Interface Agent to make highly accurate predictions. By processing a diverse range of symptoms such as cough, fever, and sore throat, the agent can feed comprehensive input into the learning model.
2. *Large Scale Data:* With over 3 million entries, the dataset provides the necessary volume for evaluating the robustness and scalability of the system. This is especially useful in a real-world scenario where the ESI Interface Agent must process a continuous influx of new COVID-19 cases.
3. *Real-Time Updates:* The incremental learning method employed by the Learner Agent ensures that the ESI Interface Agent can handle real-time data without delays. The large dataset simulates real-world conditions, where updates to the model must happen quickly to reflect new information.
4. *Binary Class Labels:* The presence of a class attribute indicating a positive or negative COVID-19 test result is crucial for the classification tasks performed by the agent. This simplifies the prediction process, allowing for straightforward binary classification and evaluation of prediction accuracy.

In summary, the "Covid Symptoms" dataset not only aligns with the core tasks of the ESI Interface Agent but also

ensures the agent can process high-volume data efficiently, update models in real time, and make accurate predictions based on symptom analysis. By incorporating this dataset into the system, the ESI Interface Agent plays a vital role in ensuring the overall success of the multi-agent COVID-19 prediction architecture.

6.3 Biskra assistant controller

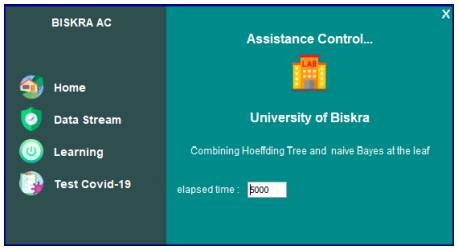


Figure 12: BISKRA interface

6.3.1 Biskra interface agent

The dataset titled 'Covid-19 Symptoms Checker,' sourced from Kaggle[38], contains 27 attributes related to Covid-19 symptoms and demographic information that may affect whether an individual has contracted the Coronavirus. Key attributes include:

- *Symptoms*: Fever, Tiredness, Dry Cough, Difficulty in Breathing, Sore Throat, and more.
- *Severity Levels*: Indicators for Mild, Moderate, Severe, and None.
- *Contact Information*: Details on contact with confirmed Covid-19 cases.
- *Demographics*: Gender (Male, Transgender) and Country.

This dataset is suitable for building a robust model to better predict Covid-19-infected patients for several reasons:

1. *Comprehensive Symptoms*: The dataset includes a wide array of Covid-19 symptoms, making it highly relevant for early detection and diagnosis in real-time applications.
2. *Severity Assessment*: Having labels for different severity levels helps in predicting the potential progression of the infection, allowing healthcare authorities to prioritize responses based on predicted outcomes.
3. *Contact History*: Information on whether individuals were in contact with confirmed cases significantly improves the predictive power of the model, aiding in identifying likely transmissions.
4. *Demographic Factors*: Attributes like gender and country allow for a detailed analysis of symptom presentation and the potential impact of demographics on infection susceptibility and disease outcomes.

6.4 The broker

The role of the Broker is to mediate between different Assistant Controllers (UK, ESI, Biskra) (Fig .13), ademonstrating the benefits of collaboration despite varying interests and goals. Autonomous and collaborative decision-making among the three Assistant Controllers is facilitated through the use of the Contract Net Interaction Protocol (Fig . 14).

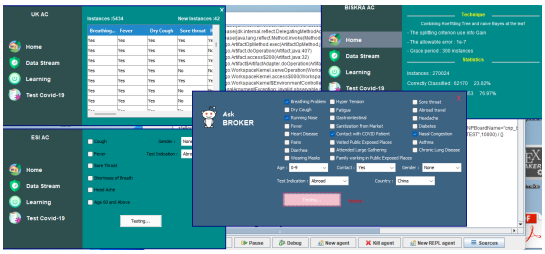


Figure 13: Broker interface

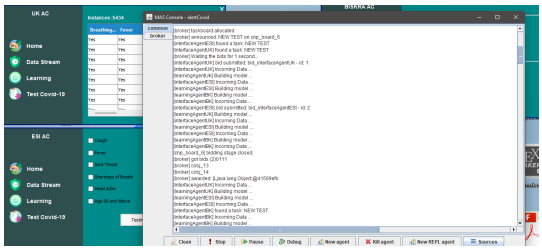


Figure 14: Broker use Contact Net protocol

7 Comparison and discussion

Fig .15 illustrates a comparison of the accuracy results from a portion of the dataset, starting from instance 7500 onward. This timeframe allows the system to build its model effectively. The comparison includes the UK, Biskra, and ESI agents, as well as the Broker Agent.

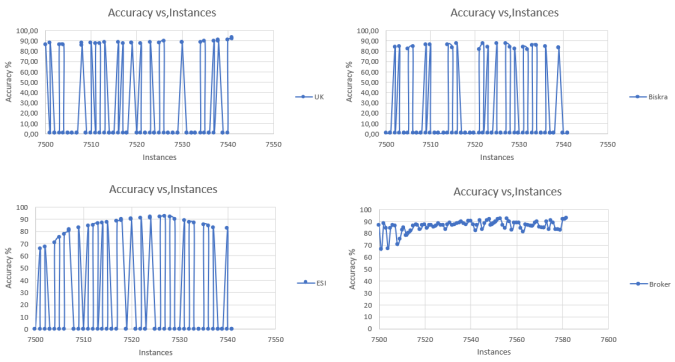


Figure 15: Broker acurracy

The performance of the UK, Biskra, and ESI ACs is summarized as follows:

- *UK AC*: Active in roughly half of the instances, with an accuracy ranging from 85% to 92% when involved.
- *Biskra AC*: Sporadically active, performing well when engaged, with an accuracy between 80% and 90%.
- *ESI AC*: The least active but consistently reliable when used, often achieving 85% or higher accuracy.

7.1 Broker performance highlights

- *Consistent Activity*: Unlike the individual agents, the broker is always active and never shows an accuracy of 0. It ensures that predictions are made even when some agents are inactive.
- *Accuracy Selection:*: The broker typically mirrors the highest-performing agent's accuracy, indicating that it selects or aggregates the best available prediction from the active agents.
- *Aggregation Role*: When multiple agents are active, the broker's performance reflects its ability to select the most accurate prediction or combine insights, ensuring higher accuracy.

7.2 Key observations

- *Broker Stability*: The broker's accuracy remains consistently strong, never dropping to 0, unlike the agents. This reliability makes the broker a crucial decision-maker, especially when certain agents are inactive.
- *Dependence on Agents*: The broker performs better when multiple agents contribute. When only one agent is active, the broker's accuracy mirrors that agent's performance. For example:
  - In instance 7541, only the UK agent contributes, resulting in the broker's accuracy being equal to the UK agent's at 92.86

The response times of the Broker Agent (in nanoseconds) across various instances are illustrated in (Fig .16). The broker's response times range from 5,600 ns to a peak of 113,700 ns in instance 7540, with another notable spike at 50,200 ns in instance 7574. These variations indicate occasional spikes in response times, possibly due to increased processing complexity or load.

The average response time of the Broker Agent is approximately 11,716 ns. Despite occasional delays, which may depend on the materials used, the broker maintains a relatively low average response time, indicating that it can efficiently manage predictions under typical conditions.

The broker plays a critical role in ensuring continuous and accurate predictions, achieving over 90% accuracy consistently over time. Utilizing the Random Forest technique, the broker consistently selects the best-performing agent's result, enhancing the overall system's reliability and performance. To further improve accuracy, increasing the participation of AC, would provide the broker with more data to work with and potentially boost overall performance.

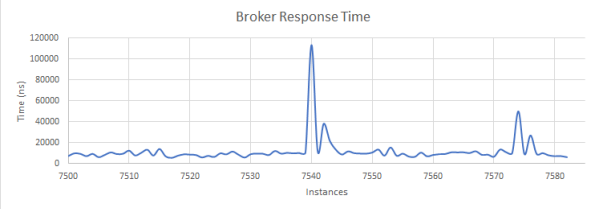


Figure 16: Broker reponse time

8 Conclusion and future work

In conclusion, this study successfully demonstrates the effectiveness of a multi-agent system (MAS) utilizing the Internet of Medical Things (IoMT) in predicting COVID-19-infected patients. The approach presented is grounded in an organizational structure within the MAS, defining the expected behaviors of agents collaborating towards a common healthcare objective. Through the implementation of experimental studies with various datasets, the proposed approach illustrates the contributions and benefits of organizational frameworks in MAS systems, achieving significant improvements in prediction accuracy, reaching over 90%.

Furthermore, we propose a flexible organizational model that allows for dynamic reorganization of the MAS in response to environmental changes. To enhance system performance, future work will focus on increasing the number of broker agents to mitigate potential bottlenecks. Additionally, rather than relying solely on random forests, we plan to explore alternative techniques for decision-making among different Assistant Controllers.

This research contributes to the growing body of knowledge in health informatics, offering viable solutions for public health authorities to monitor and respond effectively to pandemics. Future investigations could extend this model to other infectious diseases and further integrate it with existing healthcare infrastructures. The findings advocate for continued investment in smart health technologies that can transform healthcare delivery and improve patient outcomes.

References

[1] M. Gams and T. Kolenik, "Relations between electronics, artificial intelligence and information society through information society rules," *Electronics*, vol. 10, no. 4, p. 514, Feb. 2021. [Online]. Available: http://dx.doi.org/10.3390/electronics10040514

[2] S. Abderrahim and R. Maamri, "A category-theoretic approach to organization-based modeling of multi agent systems on the basis of collective phenomena and organizations in human societies," *Informatica*, vol. 42, no. 4, Apr. 2018. [Online]. Available: http://dx.doi.org/10.31449/inf.v42i4.1282

[3] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-wesley Reading, 1999, vol. 1. [Online]. Available: https://dl.acm.org/doi/10.5555/520715

[4] C. M. Toledo, R. H. Bordini, O. Chiotti, and M. R. Galli, *Developing a Knowledge Management Multi-Agent System Using JaCaMo*. Springer Berlin Heidelberg, 2012, p. 41–57. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31915-0_3

[5] H. Ahmed Abbas, "Organization of multi-agent systems: An overview," *International Journal of Intelligent Information Systems*, vol. 4, no. 3, p. 46, 2015. [Online]. Available: http://dx.doi.org/10.11648/j.ijiis.20150403.11

[6] M. H. Al-Zubaidie and R. H. Razzaq, "Maintaining security of patient data by employing private blockchain and fog computing technologies based on internet of medical things," *Informatica*, vol. 48, no. 12, Sep. 2024. [Online]. Available: http://dx.doi.org/10.31449/inf.v48i12.6047

[7] V. Janko, G. Slapničar, E. Dovgan, N. Reščič, T. Kolenik, M. Gjoreski, M. Smerkol, M. Gams, and M. Luštrek, "Machine learning for analyzing non-countermeasure factors affecting early spread of covid-19," *International Journal of Environmental Research and Public Health*, vol. 18, no. 13, p. 6750, Jun. 2021. [Online]. Available: http://dx.doi.org/10.3390/ijerph18136750

[8] J. W. Ilana Harrus, "Artificial intelligence and covid-19: Applications and impact assessment," 2021. [Online]. Available: https://www.aaas.org/ai2

[9] M. Otoom, N. Otoum, M. A. Alzubaidi, Y. Etoom, and R. Banihani, "An iot-based framework for early identification and monitoring of covid-19 cases," *Biomedical Signal Processing and Control*, vol. 62, p. 102149, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1016/j.bspc.2020.102149

[10] T. T. Nguyen, "Artificial intelligence in the battle against coronavirus (covid-19): A survey and future research directions," Aug. 2020. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.12743933.v1

[11] R. M. Carrillo-Larco and M. Castillo-Cara, "Using country-level variables to classify countries according to the number of confirmed covid-19 cases: An unsupervised machine learning approach," *Wellcome Open Research*, vol. 5, p. 56, Jun. 2020. [Online]. Available: http://dx.doi.org/10.12688/wellcomeopenres.15819.3

[12] D. Isern and A. Moreno, "A systematic literature review of agents applied in healthcare," *Journal of Medical Systems*, vol. 40, no. 2, Nov. 2015. [Online]. Available: http://dx.doi.org/10.1007/s10916-015-0376-2

[13] G. Lanzola, L. Gatti, S. Falasconi, and M. Stefanelli, "A framework for building cooperative software agents in medical applications," *Artificial Intelligence in Medicine*, vol. 16, no. 3, p. 223–249, Jul. 1999. [Online]. Available: http://dx.doi.org/10.1016/s0933-3657(99)00008-1

[14] J. M. Corchado, J. Bajo, Y. de Paz, and D. I. Tapia, "Intelligent environment for monitoring alzheimer patients, agent technology for health care," *Decision Support Systems*, vol. 44, no. 2, p. 382–396, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2007.04.008

[15] M. González Bedia, J. M. Corchado Rodríguez *et al.*, "A planning strategy based on variational calculus for deliberative agents," 2002.

[16] A. Srivastava, E.-H. S. Han, V. Singh, and V. Kumar, "Parallel formulations of decision-tree classification algorithms," in *Proceedings. 1998 International Conference on Parallel Processing (Cat. No.98EX205)*, ser. ICPP-98. IEEE Comput. Soc, p. 237–244. [Online]. Available: http://dx.doi.org/10.1109/icpp.1998.708491

[17] S. Orlando, P. Palmerini, R. Perego, and F. Silvestri, *Scheduling High Performance Data Mining Tasks on a Data Grid Environment*. Springer Berlin Heidelberg, 2002, p. 375–384. [Online]. Available: http://dx.doi.org/10.1007/3-540-45706-2_49

[18] F. Provost and V. Kolluri, "A survey of methods for scaling up inductive algorithms," *Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 131–169, 1999. [Online]. Available: http://dx.doi.org/10.1023/a:1009876119989

[19] C. Mastroianni, D. Talia, and P. Trunfio, "Managing heterogeneous resources in data mining applications on grids using xml-based metadata," in *Proceedings International Parallel and Distributed Processing Symposium*, ser. IPDPS-03. IEEE Comput. Soc, p. 11. [Online]. Available: http://dx.doi.org/10.1109/ipdps.2003.1213204

[20] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, ser. ICML '06. ACM Press, 2006, p. 161–168. [Online]. Available: http://dx.doi.org/10.1145/1143844.1143865

[21] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, p. 81–106, Mar. 1986. [Online]. Available: http://dx.doi.org/10.1007/bf00116251

[22] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM*

SIGKDD international conference on Knowledge discovery and data mining, ser. KDD00. ACM, Aug. 2000, p. 71–80. [Online]. Available: http://dx.doi.org/10.1145/347090.347107

[23] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, p. 13, Mar. 1963. [Online]. Available: http://dx.doi.org/10.2307/2282952

[24] H. Yang and S. Fong, "Incremental optimization mechanism for constructing a decision tree in data stream mining," *Mathematical Problems in Engineering*, vol. 2013, p. 1–14, 2013. [Online]. Available: http://dx.doi.org/10.1155/2013/580397

[25] A. Das, J. Wang, S. M. Gandhi, J. Lee, W. Wang, and C. Zaniolo, "Learn smart with less: Building better online decision trees with fewer training examples," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, ser. IJCAI-2019. International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, p. 2209–2215. [Online]. Available: http://dx.doi.org/10.24963/ijcai.2019/306

[26] R. Jin and G. Agrawal, "Communication and memory efficient parallel decision tree construction," in *Proceedings of the 2003 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, May 2003. [Online]. Available: http://dx.doi.org/10.1137/1.9781611972733.11

[27] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD09. ACM, Jun. 2009. [Online]. Available: http://dx.doi.org/10.1145/1557019.1557041

[28] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1137/1.9781611972771.42

[29] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, p. 1–37, Mar. 2014. [Online]. Available: http://dx.doi.org/10.1145/2523813

[30] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R news*, vol. 2, no. December, pp. 18–22, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:3093707

[31] R. G. SMITH, *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver*. Elsevier, 1988, p. 357–366. [Online]. Available: http://dx.doi.org/10.1016/b978-0-934613-63-7.50039-5

[32] R. Davis, R. G. Smith, and L. Erman, *Negotiation as a Metaphor for Distributed Problem Solving*. Elsevier, 1988, p. 333–356. [Online]. Available: http://dx.doi.org/10.1016/b978-0-934613-63-7.50038-3

[33] A. Omicini, A. Ricci, and M. Viroli, "Artifacts in the a&a meta-model for multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 3, p. 432–456, May 2008. [Online]. Available: http://dx.doi.org/10.1007/s10458-008-9053-x

[34] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi, "Multi-agent oriented programming with jacamo," *Science of Computer Programming*, vol. 78, no. 6, p. 747–761, Jun. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.scico.2011.10.004

[35] H. Hari, "Symptoms and covid presence (may 2020 data)," Aug 2020. [Online]. Available: https://www.kaggle.com/hemanthhari/symptoms-and-covid-presence

[36] W. The Departments of Computer Science, "Weka 3: Machine learning software in java," 2002. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/

[37] Nshomron, "Nshomron/covidpred: Machine learning-based prediction of covid-19 diagnosis based on symptoms." [Online]. Available: https://github.com/nshomron/covidpred

[38] B. Hungund, "Covid-19 symptoms checker," Mar 2020. [Online]. Available: https://www.kaggle.com/iamhungundji/covid19-symptoms-checker

Lagrange's Interpolation Embedded Multi-objective Genetic Algorithm to Solve Non-linear Multi-objective Optimization Problems

Muskan Kapoor[1], Bhupendra Kumar Pathak[1] and Rajiv Kumar[2]
[1]Department of Mathematics, Jaypee University of Information Technology, Solan, HP, India
[2]Department of Electronics and Communication Engineering, Jaypee University of Information Technology, Solan, HP, India
E-mail: mkapoor5628@gmail.com, pathak.maths@gmail.com, rjv.ece@gmail.com

Student paper

Keywords: Lagrange interpolation, multi-objective optimization problems, multi-objective genetic algorithm

Received: May 26, 2023

This study describes a novel strategy for solving non-linear multi-objective optimization problems encountered in real-world engineering projects. To find the best trade-off points, a Lagrange's Interpolation embedded multi-objective genetic algorithm (LI-MOGA) is used. In this approach, Lagrange's Interpolation (LI) method is used to capture the non-linear relationship between time and cost. After that, LI is combined with MOGA to create a comprehensive strategy for solving non-linear multi-objective optimization problems in the real world. The study has implications for real-time monitoring and control of the project scheduling process.

Povzetek: Predstavljen je nov pristop za reševanje nelinearnih večciljnih optimizacijskih problemov z uporabo Lagrangeove interpolacije in večciljnega genetskega algoritma (LI-MOGA), ki optimizira čas in stroške projektov.

1 Introduction

Multi-Objective Genetic Algorithm (MOGA) is an optimization technique that utilizes genetic algorithms [1] to solve problems with multiple conflicting objectives. In contrast to traditional single-objective optimization problems, MOGA considers multiple objectives simultaneously and aims to find the best possible trade-offs between them. Typically, there are multiple solutions available rather than just one, and all of these solutions are Pareto optimal [3]. MOGA is based on the principles of natural selection and genetic recombination [1], [2]. It works by evolving a population of candidate solutions to the problem, which are represented as chromosomes in a genetic algorithm. These chromosomes are evaluated based on multiple objectives, rather than a single objective, and a fitness score is assigned to each chromosome based on its performance across all objectives. The goal of MOGA is to find a set of solutions that are Pareto-optimal, meaning that no other solution in the search space is better in all objectives simultaneously. This set of solutions is known as the Pareto Front, and the process of searching for this front is known as Pareto optimization. MOGA has applications in a wide range of fields, including engineering, finance, and biology. It is particularly useful in situations where multiple objectives need to be optimized simultaneously, and where there are trade-offs between these objectives that need to be considered. Agdas et al. [4] demonstrate that the meta-heuristic approach is efficient for large-scale constructing TCT problems and that GA can be utilized to solve vast benchmark networks of variables with high levels of accuracy consistently. Genetic algorithms (GA) and linear programming (LP) are combined to create a hybrid GALP method, which is used to address optimization problems. The outcomes of the proposed model are described in [5]. Recent developments in the genetic algorithm were examined by Katoch et al. [7]. In their research paper, they examine the potential possibilities for future study in the fields of genetic operators, fitness functions, and hybrid algorithms. The competitive environment makes the relationship between time and cost even more significant, as demonstrated by Albayrak [8]. The two project components, which conflict with one another and are affected by various project constraints, must be balanced. They presented a novel hybrid algorithm (NHA) as a solution to the TCT problem, which is viewed in their paper as a multi-objective optimization problem. Likewise, several Multi-Objective Evolutionary Algorithms (MOEAs) have been suggested to find the solutions of multi-objective optimization problems. The goal of Chassiakos and Rampis' research, [9] is to examine and assess the performance potential of various evolutionary algorithms described in [9] in order to address the time-cost trade-off problems. Venkatesh et al. [10] offered a new variation of the genetic algorithm designed to solve multivariable, multi-objective, and very high search space optimization problems. Their outcomes demonstrate that this

algorithm is able to deliver positive and exact outcomes. Since it is given in real-time applications, [11] the issue of solving linear and nonlinear systems of equations is significantly beneficial and of great importance. Hassan et al. [11] introduced a comparison of several numerical approaches and GAs for solving an equation system. To distinguish between the best and worse options in order to get the best solutions, the use of genetic algorithms in numerous fields is demonstrated by Haldurai et al. [12] in their study along with how GA may be integrated with a number of other approaches to arrive at an optimal solution. Sharma et al. [13] provided a genetic algorithm-based TCT model, which was developed in a way that makes it simpler to identify the best techniques for finishing the project on schedule and for the least amount of cost. Pathak et al. [14] explained in their study how project uncertainties affect the non-linear time-cost trade-off (TCT) profile of actual engineering projects. Their study helps project managers come up with the optimal strategy for finishing a project under ambiguous conditions while also optimizing both costs and time. Most of the researchers used the evolutionary multi-objective optimization algorithms to solve linear multi-objective optimization problems like time-cost trade-off problems. In these methods, genetic operators are crucial for locating the real Pareto front.

But in non-linear MOOP, LI with evolutionary multi-objective optimization algorithms has not been investigated by researchers. This study describes an innovative and reliable approach, Lagrange's Interpolation embedded multi-objective genetic algorithm (LI-MOGA), that incorporates LI and MOGA. In this approach, LI method [15] is used to capture the non-linear relationship between time and cost. Thereafter, LI is integrated with MOGA to create an exhaustive method for solving non-linear MOOP in the real world.

2 Problem description

The non-linear relationship between project time and cost of an activity is shown in Figure 1 [6]. Suppose that a project network consists of n activities. Each activity i can be performed with (α_i) options with a corresponding cost c_i and time duration t_i. Let us assume that the vector of decision variables is taken as $X = [x_1, x_2, ..., x_i, ..., x_n]$. Now, the multi-objective TCT problem is defined by Eq. (1) and Eq. (2), respectively:

$$\text{minimize time } T = \sum_{i=1}^{n} t_i \tag{1}$$

$$\text{minimize cost } C = \sum_{i=1}^{n} c_i \tag{2}$$

where t_i and c_i can take any (α_i) options of i^{th} activity.

The project duration T is determined by computing the maximum path time (critical path method).

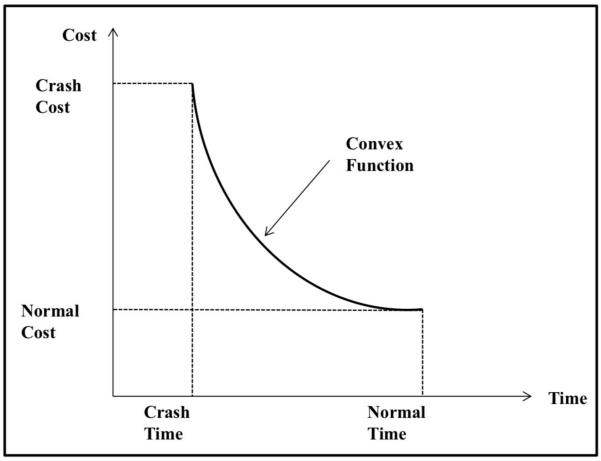


Figure 1: Non-linear time-cost trade-off relationship for an activity

3 Modelling of time-cost relationship with LI

Let's assume we have a set of input-output pairs, where the input represents the estimated completion time for a project, and the output represents the estimated cost. We can represent this set of pairs as $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$.

Using Lagrange interpolation, [17], [18] we can estimate the cost C for any given completion time T within the range of the collected input-output pairs. The Lagrange interpolation formula is:

$$\begin{aligned} f(x) = {} & \frac{(x-x_1)(x-x_2)(x-x_3)...(x-x_n)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)...(x_0-x_n)} y_0 \\ & + \frac{(x-x_0)(x-x_2)(x-x_3)...(x-x_n)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)...(x_1-x_n)} y_1 + ... \\ & + \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)...(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)(x_n-x_3)...(x_n-x_{n-1})} y_n \end{aligned} \tag{3}$$

To optimize the time and cost of the project simultaneously, we can use LI-MOGA to search for the optimal project completion time that minimizes cost while meeting the required project deadline. LI-MOGA generates a population of project activity completion times and applies the Lagrange interpolation formula to estimate the corresponding cost for each activity based on the project completion time.

4 Preliminaries of LI-MOGA

The following are the preliminary concepts of the LI-MOGA scheme for the MOOP, as used in this study:

4.1 Initial population

The initial population in LI-MOGA is typically generated randomly within the feasible search space, subject to any constraints that may be present in the problem. The initial population serves as the starting point for the optimization process and is evolved over multiple generations using genetic operators. The mathematical formulation of an initial population generated by LI-MOGA can be represented as follows:
The initial population consists of n_p solutions, where the solution is a string of type $[t_1, t_2, ..., t_i, ..., t_n]$ (Figure 2) such that $CT_i \leq t_i \leq NT_i$ and $i = 1, 2, ...n$, where CT_i and NT_i is the respective crash time and normal time of each activity.

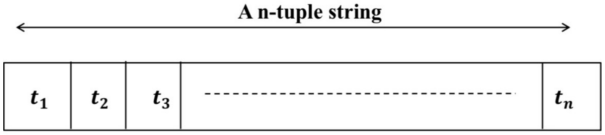


Figure 2: Time of the project schedule

The shortest time that may be used to complete an activity for a project is known as the crash time, and the expense that goes along with it is known as the crash cost and vice-versa. Each string's associated with project duration (T) and project costs (C) are determined by calculating the maximum path time and adding the costs for each activity, which is provided by LI respectively. These solutions are known as 'parents'.

4.2 Trade-off points and convex hull

In LI-MOGA, trade-off points and the convex hull of the Pareto Front are important concepts used to analyze the trade-offs between different objectives and identify the best solutions for a given problem. Trade-off points refer to the points on the Pareto Front that represent the optimal trade-offs between two objectives. Mathematically, for two objectives f_1 and f_2, the trade-off points can be defined as follows:
For a given solution x_i, let $f_i(x_i)$ be the value of the ith objective function. Then, the trade-off point $t(f_1, f_2)$ between objectives f_1 and f_2 can be defined as: $t(f_1, f_2) = (f_1(x_i), f_2(x_i))$ where x_i is a non-dominated solution that lies on the Pareto Front.

The convex hull of the Pareto front is a geometric shape that represents the set of all possible trade-offs between the objectives. The convex hull of the Pareto Front can be defined as follows:
Let N be the set of non-dominated solutions on the Pareto Front. Then, the convex hull of N can be represented as a set $CH(N)$, which contains all the extreme solutions and all the trade-off points between the objectives. Also, the extreme solution is a solution that is not dominated by any other solution in N.

4.3 Distance vs. fitness measure

The smallest distance (d_i) between the parent and each convex hull segment is determined as

$$f_i = d_{max} - d_i \tag{4}$$

$$p_i = \frac{f_i}{\sum f_i} \tag{5}$$

where p_i = probability of choosing parent i; f_i = fitness value of parent i; d_i = minimal distance between parent i and each segment j of CH; and d_i = min (d_{ij}, for all j).

4.4 Crossover operator used in LI-MOGA

Crossover is one of the important genetic operators [7] used in multi-objective genetic algorithms to generate new candidate solutions from parent solutions. The crossover operator involves exchanging genetic information between two parent solutions to create a new offspring solution.
The mathematical formulation of crossover can be represented as follows:
Let P be the population of candidate solutions, and let x_i and x_j be two parent solutions selected from P for crossover. Let n be the number of decision variables in each solution. The crossover operator involves randomly selecting a crossover point k, whereas k lies between $1 \leq k \leq n$ and exchanging the genetic information between the parents at this point. First k positions of child are taken from first k positions of x_i while remaining $(n-k)$ positions are defined by $(n-k)$ positions of x_j.
The offspring solutions are then evaluated based on their fitness using the fitness function. The offspring solutions can also undergo mutation, another genetic operator, to introduce further genetic diversity into the population as defined in the subsection 4.5.

4.5 Mutation opertor used in LI-MOGA

Mutation is a genetic operator [7] used in multi-objective genetic algorithms to introduce new genetic material into the population of candidate solutions. The mutation operator involves randomly changing the value of a decision variable in a candidate solution to create a new offspring solution.

Let P represent the population of candidate solutions, and let $x_i \in P$ be a parent solution selected for mutation, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ is a vector of n decision variables. The mutation operator involves randomly selecting a decision variable j $(1 \leq j \leq n)$ and changing its value to a new value x'_{ij} within the feasible search space. The resulting offspring solution y_i can be expressed as:

$$y_i = [x_{i1}, x_{i2}, \ldots, x_{i(j-1)}, x'_{ij}, x_{i(j+1)}, \ldots, x_{in}] \tag{6}$$

Here, x_{ij} is the original value of the j-th decision variable in the parent solution x_i, and x'_{ij} is the new value assigned to the j-th decision variable, generated within its allowable range. After mutation, the offspring solution y_i is evaluated using the fitness function. If the new solution improves the population's diversity or satisfies specific selection criteria, it may replace an existing solution in P to maintain a constant population size.

5 Working methodology of LI-MOGA

The Lagrange interpolation method is a mathematical technique for constructing a polynomial that passes through a given set of data points. Multi-objective genetic algorithm are a class of optimization algorithms that use genetic operators such as selection, crossover, and mutation to search for solutions that optimize multiple objectives simultaneously. The LI-MOGA combines these two methods to solve optimization problems.

Here, LI-MOGA scheme can be summarized as follows:

1. Define the problem: Identify the problem to be solved and the objective functions to be optimized.
2. Generate the initial population: Create an initial population of candidate solutions using random values within the problem's constraints as mentioned in subsection 4.1.
3. Select the sets of individuals i.e. non-dominating solutions known as trade-off points and draw the convex hull as given in subsection 4.2.
4. Calculate fitness: Evaluate the fitness of each candidate solution as detailed out in subsection 4.3.
5. Select parents: Use the tournament selection method to choose parent solutions for the next generation.
6. Crossover: Perform crossover on the selected parents to create new candidate solutions, refer subsection 4.4.
7. Mutation: Apply mutation to the new candidate solutions to introduce additional diversity in the population as mentioned in subsection 4.5.
8. Replace the intial population with new population.
9. Repeat: Repeat steps 3-8 until a stopping criterion is met, such as reaching a maximum number of generations or achieving a satisfactory level of convergence.

6 Benchmark problems using MOGA

Using MOGA, two common test problems Zitzler-Deb-Thiele's (ZDT1 and ZDT2) with convex Pareto-optimal front are successfully attempted [16]. It is evident from visualising the findings (as shown in graphs) that MOGA generates non-dominating solutions which are acceptable from the perspectives of diversity and convergence.

6.1 ZDT1 benchmark problem

The ZDT1 benchmark problem has two objectives which have to be minimized as illustrated below:

$$ZDT1: \begin{cases} f_1(v) & = v_1 \\ f_2(v) & = h(v)[1 - \sqrt{\frac{v_1}{h(v)}}] \\ h(v) & = 1 + \frac{9}{n-1}\sum_{i=2}^{n} v_i \end{cases}$$

This problem [16] has 30 design variables which lie in the range [0, 1] and convex Pareto-optimal solutions lies in the range $0 \leq v_1^* \leq 1$ and $v_i^* = 0$ for $i = 2, 3, ..., 30$. According to Figure 3, the non-dominated solutions obtained using the MOGA technique are well dispersed throughout the solution space and matches the Pareto-optimal front in a reasonable way. Thus, the MOGA technique proves its capacity for convergence to the Pareto-optimal front and for discovering a variety of solutions for complex situations.

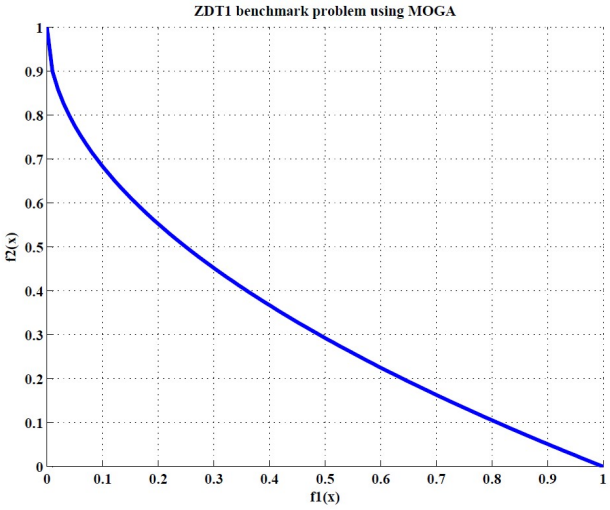


Figure 3: ZDT1 benchmark problem using MOGA

6.2 ZDT2 benchmark problem

The ZDT2 function has a non-convex Pareto-optimal front. The objective functionas are as mentioned below;

$$ZDT2: \begin{cases} f_1(v) & = v_1 \\ f_2(v) & = h(v)[1 - (\frac{v_1}{h(v)})^2] \\ h(v) & = 1 + \frac{9(\sum_{i=2}^{n} v_i)}{n-1} \end{cases}$$

Thirty design variables v_i ; $i = 1, 2, ..., n$ were selected for this ZDT2 function $(n = 30)$. Every design variable had a value between 0 and 1. When $h = 1.0$, the Pareto-optimal

front arises [16]. The Pareto-optimal front is where the non-dominated solutions from the MOGA technique reside, and the solutions are well dispersed in the solution space (Figure 4). This demonstrates how the MOGA technique for this test problem converges to the Pareto-optimal front.

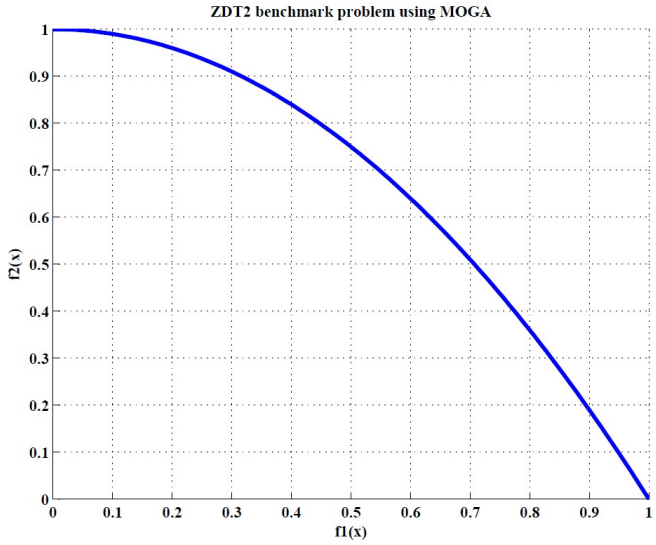


Figure 4: ZDT2 benchmark problem using MOGA

7 Results and discussion

As a test problem, the real-world project network shown in Figure 5 [3] is appropriately modified to include a non-linear relationship between activity time and cost. The problem is then tackled using the proposed approach (LI-MOGA). Table 1 also displays the various aspects of each activity.

Table 1: Alternatives of test problem

ID	Time	Cost	ID	Time	Cost
1	14	2400	10	15	450
1	15	2150	10	22	400
1	16	1900	10	23	390
1	18	1750	10	27	345
1	21	1500	10	28	330
1	23	1300	10	30	325
1	24	1200	10	33	320
2	15	3000	11	12	450
2	17	2630	11	13	420
2	18	2400	11	14	370
2	20	1800	11	16	350
2	21	1720	11	17	305
2	23	1500	11	19	300
2	25	1000	11	20	300
3	15	4500	12	22	2000
3	17	4415	12	24	1750
3	19	4220	12	25	1690
3	22	4000	12	27	1525
3	25	3730	12	28	1500
3	30	3375	12	29	1200
3	33	3200	12	30	1000
4	12	45000	13	14	4000
4	13	44300	13	15	3795
4	15	38450	13	16	3500
4	16	35000	13	18	3200
4	18	33700	13	21	2750
4	19	32400	13	23	2155
4	20	30000	13	24	1800
5	22	20000	14	9	3000
5	24	17500	14	10	2930
5	25	16400	14	12	2825
5	26	15900	14	14	2605
5	27	15700	14	15	2400
5	28	15000	14	17	2295
5	30	10000	14	18	2200
6	14	40000	15	10	6525
6	16	39200	15	13	5990
6	17	34500	15	14	4500
6	18	32000	15	16	3500
6	20	27700	15	17	3355
6	22	20300	15	18	2600
6	24	18000	15	20	1930
7	9	30000	16	20	3000
7	11	27200	16	22	2000
7	13	26100	16	24	1750
7	14	25600	16	26	1685
7	15	24000	16	28	1500
7	17	22300	16	29	1385
7	18	22000	16	30	1000
8	14	220	17	14	4000
8	15	215	17	16	3700
8	16	200	17	17	3455
8	17	190	17	18	3200
8	21	167	17	21	2780
8	23	150	17	23	2335
8	24	120	17	24	1800
9	15	300	18	9	3000
9	16	280	18	10	2900
9	18	245	18	12	2790
9	21	175	18	14	2565
9	22	135	18	15	2400
9	24	115	18	16	2315
9	25	100	18	18	2200

Continued on next page

Continued from previous page

Initial population (n_p), mutation rate (p_m), and crossover rate (c_r) are defined, respectively, as 540, 0.02, and 1.0. Additionally, as 5 consecutive iterations were determined to be a sufficient amount, the search is configured to end when the trade-off profile does not change during

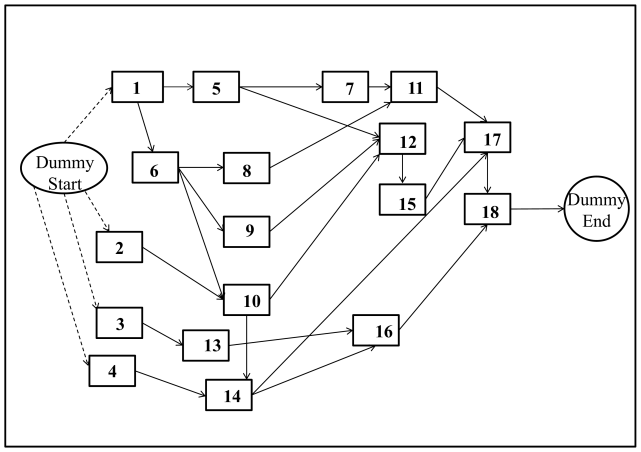


Figure 5: Network of test problem

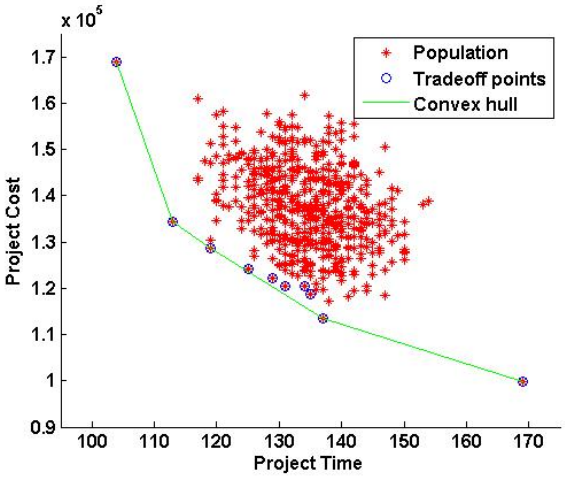


Figure 6: Initial generation

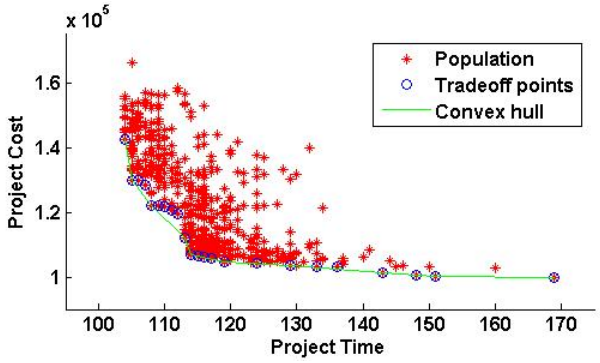


Figure 7: Intermediate generation

that time. Initial population n_p of 540 strings is randomly selected. An exhaustive experiment is used to select an initial population, and 540 is ultimately determined to be a suitable number. Because the population's diversity is preserved and the convergence period is short, the population is scattered across all feasible spaces. Figure 6 illustrates how n_p is dispersed throughout the solution space and does not concentrate in any particular area. The new population shifts in the direction of axes with intermediate improvements in trade-off profile and convex hull are shown in Figure 7. A clear improvement in trade-off points and convex hull across generations can be seen in the best-attained trade-off points and their convex hull (Figure 8). This profile is determined to be the best TCT profile according to LI-MOGA search criteria because the trade-off profile does not improve further.

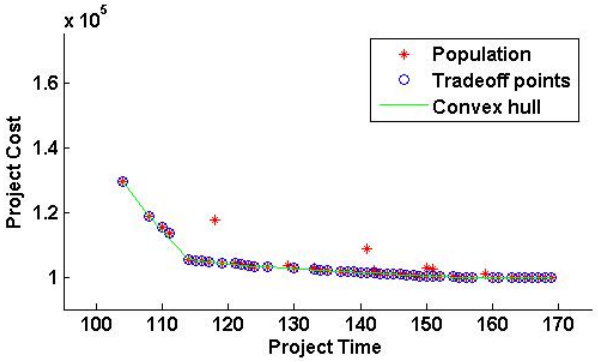


Figure 8: Final generation

8 Conclusion

LI-MOGA can identify optimal activity allocation using any time-cost functions, and it is not constrained by the form of the activity time-cost relationship. In order to validate the performance of the MOGA utilized, two common test problems are effectively addressed. LI's versatility and higher modeling capacity are demonstrated by providing various cost-evaluation options for each activity. The integrated approach uses a novel combination of LI and MOGA to address non-linear multi-objective optimization problems. This is a powerful and efficient multi-objective optimization technique that can search for a near-optimal/optimal realistic TCT profile after exploring a small search space. The proposed approach can assist in real-time to choose the best alternative among non-linear TCT profiles to execute the real world projects.

References

[1] John H. Holland. Adaptation in natural and artificial systems. Ann Arbor: The University of Michigan Press, Michigan, pp. 1–23, 1975. https://doi.org/10.1145/1216504.1216510

[2] David E. Goldberg. Genetic algorithms in search, optimization, and machine learn-

ing. Addison-Wesley, Reading, p. 673, 1989. https://doi.org/10.5860/choice.27-0936

[3] Chung-Wei Feng, Liang Liu, and Scott A. Burns. Using genetic algorithms to solve construction time-cost trade-off problems. Journal of Computing in Civil Engineering, 11(3):184–189, 1997. https://doi.org/10.1061/(asce)0887-3801(1997)11:3(184)

[4] Duzgun Agdas, David J. Warne, Jorge Osio-Norgaard, Forrest J. Masters. Utility of genetic algorithms for solving large-scale construction time-cost trade-off problems. Journal of Computing in Civil Engineering, 32(1), 2018. https://doi.org/10.1061/(asce)cp.1943-5487.0000718

[5] S.M. Reza Alavipour and David Arditi. Time-cost tradeoff analysis with minimized project financing cost. Automation in Construction, 98:110–121, 2019. https://doi.org/10.1016/j.autcon.2018.09.009

[6] Bhupendra Kumar Pathak, Sanjay Srivastava, and Kamal Srivastava. Neural network embedded multiobjective genetic algorithm to solve non-linear time-cost tradeoff problems of project scheduling. Journal of Scientific and Industrial Research, 67, 2008.

[7] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80:8091–8126, 2021. https://doi.org/10.1007/s11042-020-10139-6

[8] Gülçağ Albayrak. Novel hybrid method in time–cost trade-off for resource-constrained construction projects. Iranian Journal of Science and Technology, Transactions of Civil Engineering, 44(4):1295–1307, 2020. https://doi.org/10.1007/s40996-020-00437-2

[9] A.P. Chassiakos and G. Rempis. Evolutionary algorithm performance evaluation in project time-cost optimization. Journal of Soft Computing in Civil Engineering, 3(2):16–29, 2019.

[10] S.S. Venkatesh and Deepak Mishra. Variable search space converging genetic algorithm for solving system of non-linear equations. Journal of Intelligent Systems, 30(1):142–164, 2020. https://doi.org/10.1515/jisys-2019-0233

[11] Osama Farouk Hassan, Amani Jamal, Sayed Abdel-Khalek. Genetic algorithm and numerical methods for solving linear and nonlinear system of equations: a comparative study. Journal of Intelligent & Fuzzy Systems, 38(3):2867–2872, 2020. https://doi.org/10.3233/jifs-179572

[12] L. Haldurai, T. Madhubala, and R. Rajalakshmi. A study on genetic algorithm and its applications. International Journal of Computer Sciences and Engineering, 4(10):139, 2016.

[13] Deepak Sharma and Sanjay Tiwari. Genetic algorithm-based optimization model for time-cost trade-off for construction project. International Journal for Research in Applied Science & Engineering Technology, 2023. https://doi.org/10.22214/ijraset.2023.48621

[14] Bhupendra Kumar Pathak and Sanjay Srivastava. Effects of project uncertainties on nonlinear time-cost tradeoff profile. Iranian Journal of Fuzzy Systems, 12(4):79–100, 2015.

[15] E.T. Whittaker and G. Robinson. The calculus of observations: a treatise on numerical mathematics. Journal of the Royal Statistical Society, 87(2):291–293, 1924. https://doi.org/10.2307/2341229

[16] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, 8(2):173–195, 2000. https://doi.org/10.1162/106365600568202

[17] H. Jeffreys and B.S. Jeffreys. Methods of mathematical physics. Cambridge University Press, 1999. https://doi.org/10.1017/cbo9781139168489

[18] R. Bridges. Programming for mathematicians. The Mathematical Gazette, 2000. https://doi.org/10.2307/3620810

CONTENTS OF Informatica Volume 48 (2024) pp. 1-712

Papers

HAMLA, H. & K. GHANEM. 2024. A Hybrid Feature Selection Based on Fisher Score and SVM-RFE for Microarray Data. Informatica 48: 57-68.

HIEN, N.L.H. & L.V. HUY, H.H. MANH, N.V. HIEU. 2024. A Deep Learning Model for Context Understanding in Recommendation Systems. Informatica 48: 31-44.

HROVATIN, N. 2024. Enabling Decentralized Privacy Preserving Data Processing in Sensor Networks. Informatica 48: 141-142.

JOSE, J. & V. MATHEW. 2024. Internet of Things – A Model for Data Analytics of KPI Platform in Continuous Process Industry. Informatica 48: 119-130.

KAPOOR, M. & B.K. PATHAK, R. KUMAR. 2024. Lagrange's Interpolation Embedded Multi-Objective Genetic Algorithm To Solve Non-linear Multi-objective Optimization Problems. Informatica 48: 699-706.

KAR, C. 2024. Tropical Cyclone Intensity Estimation by Feature Extraction Techniques Using Satellite Imagery. Informatica 48: 497-498.

LATRECHE, I. & S. SLATNIA, O. KAZAR, E. BARKA, S. HAROUS. 2024. A Review on Deep Learning Techniques for EEG-Based Driver Drowsiness Detection Systems. Informatica 48: 359-378.

LI, Z. 2024. The Application of Multiple Regression Model in Blended Teaching of Higher Mathematics. Informatica 48: 473-484.

M, P. & S. PATIL, M.M. V, S.M. S, V. K. 2024. CerConvNet: Cervical Cancer Cells Prediction Using Convolutional Neural Networks. Informatica 48: 439-454.

MAHMED, A.N. & M.N.M. KAHAR. 2024. Simulation for Dynamic Patients Scheduling Based on Many Objective Optimization and Coordinator. Informatica 48: 91-106.

MERAD, B. & K. BELKADI. 2024. O-AIRS: Optimized Artificial Immune Recognition System. Informatica 48: 347-358.

MOALLA, H. & H. ABID, D. SALLAMI, E. AÏMEUR, B.B. HAMED. 2024. Exploring the Power of Dual Deep Learning for Fake News Detection. Informatica 48: 567-594.

OUF, S. & M. ASHRAF, M. ROUSHDY. 2024. A Proposed Paradigm Using Data Mining to Minimize Online Money Laundering. Informatica 48: 309-330.

PALANIAPPAN, R. 2024. An Overview on Robot Process Automation: Advancements, Design Standards, its Application, and Limitations. Informatica 48: 1-10.

PATIL, D.R. 2024. A Framework for Malicious Domain Names Detection using Feature Selection and Majority Voting Approach. Informatica 48: 419-438.

PORANKI, V.K.R. & B.S. RAO. 2024. DRG-Net: Diabetic Retinopathy Grading Network Using Graph Learning with Extreme Gradient Boosting Classifier. Informatica 48: 171-184.

RAJPUT, B.S. & P. ROY, S. SONI, B.S. RAGHUWANSHI. 2024. ELM-Based Imbalanced Data Classification-A Review. Informatica 48: 185-204.

RAMAIJANE, T. & H. HLOMANI, I. ZLOTNIKOVA, T. MAUPONG. 2024. Evaluation of Manifold Dual Contouring Algorithms Based on k-d tree and Octree Data Structures. Informatica 48: 399-418.

SAFIR, S. & A. SIAM. 2024. Predicting Covid-19 Infections with a Multi-Agent Organizational Approach and Machine Learning Techniques. Informatica 48: 685-698.

SAHAKYAN, H. & G. KATONA, L. ASLANYAN. 2024. Study on Using Reinforcement Learning for the Monotone Boolean Reconstruction. Informatica 48: 521-532.

SAHU, R. & S.R. DASH, A. BARAL. 2024. Identification of Students' Confusion in Classes from EEG Signals using Convolution Neural Network. Informatica 48: 45-56.

SHIN, J. 2024. Deep Learning Assisted Intelligent Human Computer Interaction for Next Generation Internet Applications. Informatica 48: 145-146.

SINGAL, P. & P. SHARMA, A.C. KUMARI. 2024. Integrated Software Effort Estimation: a Hybrid Approach. Informatica 48: 331-346.

SLAPNIČAR, G. 2024. Contact-Free Physiological Monitoring of Cardiorespiratory States Using Radar and Optical Sensors. Informatica 48: 499-502.

SREE, V.P. & A.J. DAVULURI, S. POLEPALLI. 2024. Classification and Identification of Weeds Using Machine Learning Classifiers. Informatica 48: 653-662.

SRINIVAS, K. & V. CH, S.B. BORRA, K.S. RAJU, G.R.K. RAO, K.N.V. SATYANARAYANA, P.M. KUMAR. 2024. Cardiacnet: Cardiac Arrhythmia

Detection and Classification Using Unsupervised Learning Based Optimal Feature Selection with Custom CNN Model. Informatica 48: 147-158.

STOJKOVIKJ, N. & L.K. LAZAROVA, A. STOJANOVA, M. MITEVA, B. ZLATANOVSKA, M. KOCALEVA. 2024. Application of Agent-Based Modelling in Learning Process. Informatica 48: 11-20.

VASHISHTHA, S. & D. VARSHNEY, E. SARIN, S. KAUR. 2024. A Novel Music Recommendation System Using Filtering Techniques. Informatica 48: 595-610.

VLAHEK, D. 2024. An Efficient Iterative Algorithm to Explainable Feature Learning. Informatica 48: 289-290.

WANG, X. 2024. Edge Computing Based Multi-Objective Task Scheduling Strategy for UAV with Limited Airborne Resources. Informatica 48: 255-268.

WU, M. & Z. LIU. 2024. Unmanned Logistics Vehicle Control based on Path Tracking Control Algorithm. Informatica 48: 239-254.

XU, X. 2024. Multimedia VR Image Improvement and Simulation Analysis Based on Visual VR Restructuring Algorithm. Informatica 48: 107-118.

ZHANG, J. 2024. Attribute Induction-oriented Excavation and Generalization Analysis of Site Archaeological Data. Informatica 48: 223-238.

ZHAO, H. 2024. Animation Character Mouth Matching Model Considering Reinforcement Learning and Feature Extraction. Informatica 48: 455-472.

ÁVILA-DÍAZ, M. & M. MÁRQUEZ-VERA, O. DÍAZ-PARRA, V. PUIG, A. MA'ARIF. 2024. Inverse Fuzzy Fault Models for Fault Isolation and Severity Estimation in Industrial Pneumatic Valves. Informatica 48: 379-398.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several sci- entific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute tem- perature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, en- ergy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research de- partments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the uni- versities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; ap- plied mathematics. Most of the activities are more or less closely connected to information sciences, in particu- lar computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automa- tion and control, professional electronics, digital communi- cations and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the in dependent state of S**love**nia (or S♡nia). The capital today isconsidered a crossroad bet between East, West and Mediter-ranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strat- egy for technological development to foster synergies be- tween research and industry, to promote joint ventures be- tween university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.: +386 1 4773 900, Fax.:+386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2022 (Volume 46) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Blaž Mahnič, Gašper Slapničar; gasper.slapnicar@ijs.si
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Slovene Society for Pattern Recognition (Vitomir Štruc)
Slovenian Artificial Intelligence Society (Sašo Džeroski)
Cognitive Science Society (Olga Markič)
Slovenian Society of Mathematicians, Physicists and Astronomers (Dragan Mihailović)
Automatic Control Society of Slovenia (Giovanni Godena)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Mark Pleško)
ACM Slovenia (Ljupčo Todorovski)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 48 Number 4 December 2024 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **48** (2024) Number 4, pp. 505–712