Volume 48 Number 3 September 2024 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

1977

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.

A New Version of a Broadly Applicable, Cross-lingual Meaning Representation Formalism and Its Significance for Biomedical Sciences

Vladimir A. Fomichov
Department of Systems Modeling and Design Automation, Moscow Aviation Institute (National Research University)
Moscow, Russian Federation
E-mail: vfomichov@gmail.com

Keywords: automatic information extraction, natural language processing, semantic parsing of scientific definitions, scholarly knowledge, biomedical sciences, abstract meaning representation, uniform meaning representation, theory of K-representations, SK-language

Received: March 10, 2024

The first purpose of the paper is to attract the attention of the scholars in natural language (NL) processing to a new version of a broadly applicable, cross-lingual meaning representation formalism: a new version of the theory of SK-languages (standard knowledge languages) introduced in early 2000s by the theory of K-representations (knowledge representations), or TKR. A collection of original expressive mechanisms for constructing semantic representations (SRs) of scientific notions' definitions is suggested, its joint work is demonstrated. A special attention is paid to indicating the advantages of TKR-based approach to building SRs of scientific definitions in comparison with Universal Conceptual Cognitive Annotation, Abstract Meaning Representation, and Uniform Meaning Representation. New precious and broad prospects of describing semantic structure of NL-texts pertaining to biomedical sciences are indicated. The second purpose is to improve an algorithm (introduced in a previous paper of the author) constructing SRs of scientific notions' definitions and to illustrate its principal ideas. The output SRs are the expressions of SK-languages. The methodological basis is TKR. In particular, the suggested algorithm of scientific definitions' semantic parsing includes a complex procedure based on the algorithm of semantic parsing SemSynt1 given by TKR. The original features of the suggested algorithm constructing SRs of scientific definitions are shown.

Povzetek: Članek uvaja izboljšano različico SK-jezikov za medjezikovno reprezentacijo pomena v obdelavi naravnega jezika, s poudarkom na biomedicinskih znanostih, ter izboljšuje algoritme za semantično razčlenjevanje znanstvenih definicij.

1 Introduction

During two last decades, a considerable progress has been achieved in many branches of the field of studies aimed at processing written texts and oral speech in natural language (NL). In particular, it applies to NL-interfaces to mobile devices, OWL-based ontologies, autonomous intelligent systems (robots), the computer systems extracting knowledge from NL-texts with the goal of forming and updating ontologies, question-answering systems dealing with full-text databases, NL-interfaces to the giant semantic information system Linked Open Data (LOD).

In the context of this progress, many specialists in NL processing (NLP) from various countries have realized the existence of a fundamental problem to be solved in order to continue this progress. This fundamental problem is the construction of broadly applicable (desirably, universal), cross-lingual meaning representation (MR) formalisms and of the algorithms transforming NL-texts into their meaning representations (or semantic representations, SRs) with respect to the used knowledge base and, in the case of discourses, in the context of the previous fragment of the analyzed discourse.

The recent impressive achievements in constructing NLP systems based on large languages models and black-box neural networks could stimulate many computer science specialists to draw the conclusion that the problem of developing broadly applicable, cross-lingual MR formalisms is not worth to concentrate much attention on solving it. However, the authors of [1, 2] underline that the black box nature of neural networks makes it difficult to know when and where to correct the used models for eliminating errors or at least anticipating errors. But it is absolutely necessary to have the possibilities of the kind, because interpretability and controllability in NLP systems are critical in high-stake application scenarios, in particular, for realizing reasonable human-intelligent robot interaction and in medicine for making reasonable expectations and conclusions.

The above said explains the reasons for organizing the First International Workshop on Designing Meaning Representations (DMR 2019, Florence, Italy, August 2019) [3]. The principal objectives of the workshop DMR 2019 were as follows: (a) to gain a deeper understanding of the key elements of meaning representations (MRs) that are the most valuable to the NLP community; (b) to critically examine existing MRs with the goal of using the findings to inform the design of next-generation MRs; (c) to explore the opportunities and identify the challenges in the design and use of MRs in multilingual settings [3]. The Fifth International Workshop on Designing Meaning Representations (DMR 2024) took place in May 2024 in Torino, Italy [4].

The proceedings of the workshop DMR 2019 include the paper [5], its authors (three specialists from two IBM research centres in San Jose, California) underline that "creating a universal semantic representation that works across a large number of languages is an important objective for the NLP community".

Wonderfully, the first version of mathematical framework being appropriate for creating a universal, cross-lingual semantic representation was suggested 23 years before the start of the workshop DMR 2019 in the paper [6]. This paper introduced a mathematical model (having an original form) describing a system of ten partial operations on conceptual structures and introducing a new class of formal languages – the class of restricted standard knowledge languages (RSK-languages).

The paper [7] shortly explains the essence of these ten partial operations on conceptual structures. This information is used in the mentioned paper as the ground for demonstrating very broad expressive possibilities of RSK-languages. In particular, the possibilities of building SRs of compound scientific notions and of complex discourses. A considerable part of the paper [7] is devoted to explicating the advantages of the RSK-languages in comparison with Universal Networking Language (UNL) elaborated under the guidance of the UNO Institute for Advanced Studies, Tokyo University. This language is based on the idea of representing the meanings of separate sentences by means of binary relations.

The second version of a mathematical framework being appropriate for creating a universal, cross-lingual MR was published nine years before the workshop DMR 2019. It was done in the monograph [8] introducing the theory of K-representations (knowledge representations) – an original theory of designing semantic-syntactic parsers of NL with the broad use of formal means for representing input, intermediary, and output data. One of the constituents of the theory of K-representations (TKR) is the theory of SK-languages (standard knowledge languages).

The difference between the theory of RSK-languages and the theory of SK-languages can be explained as follows. The first theory assumes the existence of only one possible angle of look at any considered entity from an application domain. For instance, people are considered as intelligent objects, the cars – as dynamic physical objects, and the firms – as the organizations. However, each concrete person is simultaneously both an intelligent object and a dynamic physical object, the IT-companies develop, in particular, programming environments and have the locations. That is why the theory of SK-languages gives the possibility to consider the compound semantic characteristics (compound types) of the entities from application domains. For instance, each person may be associated with the type *intel. system * dyn. phys. object* and each firm with the type *org * intel. system * space. object*.

The essence of the principal constituents of TKR is shortly explained below in the sections 3 and 4.

Taking into account the objectives of the present paper, it is important to say that the monograph [8] includes the following conjecture: "The designers of NL processing systems have received a system of the rules for constructing well-formed formulas (besides, a compact system, it consists of only ten main rules) allowing for (according to the hypothesis of the author) building semantic representations (SRs) of arbitrary texts pertaining to numerous fields of humans' professional activity, i.e., SRs of the NL-texts on economy, medicine, law, technology, politics, etc." [8, p. ix-x].

Since the release of the monograph [8], no scholars have put forward any objections against this hypothesis.

There are serios reasons to believe that the construction of the theory of SK-languages in [8] means that the barrier of complexity in developing a universal meaning representation formalism was overcame nine years before the workshop DMR 2019 and, as it is clear now, outstripped the time of its creation.

The first objective of the present paper is to attract the attention of the NLP community to a new version of the SK-languages' theory and, as a consequence, of TKR being more compact and convenient for practical usage in comparison with the version of TKR stated in the monograph [8]. A new version is constructed due to introducing a compact mathematical model of a system of primary units of conceptual level used by an applied intelligent system (AIS). Formally, this goal is achieved due to introducing and considering the notion of *optimized conceptual basis* instead of the notion of conceptual basis.

The present paper explicates the broadest prospects opened for biomedical sciences by SK-languages. Two principal directions of reasoning are combined: (a) the demonstration of the possibilities to use SK-languages for building SRs of complex scholarly discourses and complex definitions of the notions; (b) the explication of the advantages of SK-languages in comparison with Universal Conceptual Cognitive Annotation [9], Abstract Meaning Representation [10], and Uniform Meaning Representation [1].

The second objective of the paper is to improve the algorithm AlgSemDef1 introduced in [11] constructing SRs of scientific notions' definitions and to illustrate the principal ideas of the suggested algorithm AlgSemDef2. The output SRs of the definitions are the expressions of SK-languages.

In Conclusions, the following hypothesis is formulated: the considered optimized version of the SK-languages' theory (a part of TKR) may be interpreted as the *starting universal meaning representation formalism.*

2 Computational semantics and biomedical sciences

During last decade, the development of the huge semantic information system Linked Open Data (LOD) [12, 13] caused the emergence of a big family of projects in the field of NL processing aimed at extracting factual information in the form of triples. Mainly due to this reason, the attention of the major part of researchers in NLP was not focused on the problem of representing semantic structure of scientific texts in NL (the NL-texts) in a formal way.

A strong impulse for "switching-on" the interest of the researchers in NLP to this problem was given in the middle of the 2010s by the change of the paradigm in the field of constructing large knowledge bases (KBs) in biomedical sciences. The first limitation of the previous period of constructing biomedical KBs is that, most often, they were manually built and curated. The second limitation was that the prior work on automatic information extraction (AIE) from NL-texts was that it was focused on several distinguished classes of scientific publications describing, in particular, protein–protein interactions or gene-drug relationships or drug effects, etc.

The main aspects of changing (in the middle of the 2010s) the paradigm of AIE for constructing biomedical KBs is the transition to considering, as knowledge sources, not only arbitrary scientific publications in the field but also the sources like the health portals and popular online discussion forums. This transition is considered as the ground for developing new, comprehensive approaches linking diverse entity types, spanning genes, diseases, symptoms, drugs, drug effects, anatomic parts, etc. [14].

As a consequence, the designers of computer systems for constructing large knowledge graphs for biomedical sciences faced the need of strong and flexible formal means for representing semantic structure of complex NL-expressions.

During last fifteen years, the researchers working in NL processing have received four main formalisms for designing semantics-oriented NLP systems extracting knowledge from biomedical texts. The first one was suggested by V.A. Fomichov in the invited keynote address delivered at the opening session of the 18th international TALN conference (Traitement Automatique des Langues Naturelles, June 27 – July 1, 2011, France, University Montpellier 2) [15]. The principal subject of the keynote address was the broad prospects opened for bioinformatics and Semantic Web by the theory of K-representations (TKR) stated in [8].

Taking into account the objectives of the present paper, it should be noted that the keynote address [15] attracted, in particular, the attention of the researchers in bioinformatics to the expressive mechanisms being precious for representing scholarly knowledge, first of all, the definitions of scientific notions.

The approach UCCA [9] is qualified by its authors as a semantic representation. However, it seems that it may be qualified more exactly as a kind of semantic-syntactic representation (SSR). The reason for this conclusion is that the basic elements of the UCCA structures are not semantic units but the words (including the articles "a", "the', the pronouns "he", "his", etc.) and word combinations. The UCCA approach suggests to use directed acyclic graphs as SSRs of the sentences in NL. The authors of [9] suggested 12 original labels for marking the vertices of the graph. The basic expressions are called *scenes*, they may be of two types: *processual scenes (PS)* and *state scenes (StS).*

One of the edges starting from the root of PS has the label P (Process), and one of the edges starting from the root of StS has the label S (State). The label A (Participant) denotes a participant in a scene in a broad sense. E.g., a linear representation of a scene associated with the sentence "Cukor encouraged the studio to accept her demands" is the expression

$$\textit{Cukor}_A\ \textit{encouraged}_P\ [\textit{the}_E\ \textit{studio}_C]_A\ [\textit{to}_R\ [\textit{accept her demands}]_C\ ]_A.$$

Here the label "C" (Center) is used for the conceptualization of the parent unit, "E" (Elaborator) marks a non-scene relation which applies to a single Center, "R" (Relation) corresponds to all other types of non-scene relations.

The main drawbacks of the UCCA approach seem to be as follows: (a) the basic elements are the words but not semantic units; (b) there are no expressive mechanisms for constructing SRs of the expressions in NL listed in the Table 1.

The semantic formalism AMR was introduced in the year 2013 in the ACL publication [10] by a group consisting of ten researchers from UK and USA. The central idea of AMR approach is to use rooted, directed acyclic graphs for representing the meaning of a sentence. The nodes in the graphs (called concepts) stem from the words and word combinations in a sentence, and the edges correspond to semantic-syntactic relationships between the words in the sentence. During last decade, a number of research groups in different countries realized the experimental projects aimed at parsing a sentence into its AMR [16-20].

The subject of the paper [21] is biomedical events extraction using AMR, the events are protein-protein interactions. An example of such sentence is as follows: "This LRA-induced rapid phosphorylatin of radixin was significantly suppressed in the presence of C3 toxi, a potent inhibitor of Rho".

The most recent specification of AMR is dated by May 1, 2019 [22].

During mainly last seven years, a number of the projects have emerged aimed at investigating the possibilities of using AMR for representing the meanings of sentences in Chinese, Turkish, Korean, Portuguese, Spanish, Vietnamese. The common feature of the publications describing such projects is that the researchers have considered only rather simple sentences of concrete languages being structurally very far from complex sentences encountered in scholarly documents.

AMR considers only sentences but not the discourses (the sequences of sentences interrelated by their meanings). For overcoming this restriction, the Uniform Meaning Representation was suggested in [1]. UMR consists of an AMR-based sentence level representation of a text that focuses on predicate-argument structures, the senses and named entities and a document level representation (in other words, a discourse level representation) that captures semantic relations going beyond the sentence boundaries. It applies to the time and causal relations and also to the coreference relations establishing the identity of several designations of the same entity.

At the sentence level, UMR adds to named entities and words senses that are already in AMR the means to express the information of syntactic character. E.g., a sentence level UMR for the sentence “He denied any wrong-doing” associates the pronoun “he” with the concept *“a person”* that has a *ref-person* attribute with the value *3rd* and a *ref-numb* attribute with the value *Singular*. That is why, as in case of UCCA, UMR may be better qualified as semantic-syntactic representation.

The paper [23] describes an UMR-writer – a Web-based software helping to created te UMR annotations of NL-texts. The paper [24] reports the first release of the UMR dataset consisting of six languages – Chinese, English, Arapaho, Kukavra, Navajo, and Sanapana where the last four are low resources languages that have quite distinct linguistic properties.

The focus of this section is on the significance of main modern branches of formal NL semantics for designing computer systems being able to extract knowledge from the sentences and discourses pertaining to biomedical sciences. In this connection, let’s formulate several properties of semantic formalisms seeming to be crucial for the design of NLP systems extracting knowledge from biomedical NL-texts.

Property P1. The possibility to form compound designations of the notions (“an enzyme for converting fibrinogen to fibrin during coagulation”, etc.). It should be mentioned that the creators of the system KnowLife indicated in [14] the necessity of representing compound notions.

Property P2. The possibility to construct a definition of a scientific notion connecting an introduced notion with its explanation. For instance, it is the case of the definition D1 = “The genotype of an organism is the collection of all its chromosomes”.

Property P3. The availability of effective formal means for building SRs of arbitrarily complex descriptions of the sets. As an example, let’s consider the definition D2 = “Type A blood group are the persons who possess type A isoantigen on red blood cells and anti-B agglutinin in plasma”.

Property P4. The possibility to build SRs of the infinitives with dependent words or gerundial constructions. Such expressions may express the goals, commitments, the destinations of the processes, wishes, etc. As a example, let’s consider the expression “an enzyme which helps to convert fibrinogen to fibrin during coagulation” as a fragment of the definition D3 = “Thrombin is an enzyme which helps to convert fibrinogen to fibrin during coagulation”.

Property P5. The possibility to construct SRs of the compound constructions formed from several infinitives with dependent words with the help of the logical connectives NOT, AND, OR.

Property P6. The possibility to build SRs of complex sentences and discourses with the references to the meanings of phrases and larger parts of discourse. For example, this phenomenon is realized in the definition D4 = “All granulocytes are polymorphonuclear; that is, they have multilobed nuclei”.

Property P7. The availability of formal means allowing for constructing object-oriented SRs of definitions, that is, formal structures including such slots as the list of the authors of a definition, the year of its creation, a relevant thematic domain, and SR of the definition.

This property is significant for the process of inscribing the constructed SR of a definition into an appropriate ontology.

Table 1 shows whether the approaches UCCA, AMR, UMR, TKR possess the listed properties 1 – 7.

Table 1. The possession of the properties P1 – P7 by the approaches UCCA, AMR, UMR, TKR.

Pro-perty	UCCA	AMR	UMR	TKR
P1	-	-	-	+
P2	-	-	-	+
P3	-	-	-	+
P4	-	-	-	+
P5	-	-	-	+
P6	-	-	-	+
P7	-	-	-	+

The present paper continues the line of the paper [25], where it was shown that much broader prospects in comparison with AMR for creating semantic languages – intermediaries are opened by TKR. The content of Table 1 shows that this conclusion applies also to the approaches UCCA and UMR.

One of the objectives of the present paper is to remind the designers of the systems extracting information from scientific texts for constructing large biomedical knowledge graphs of the powerful and flexible expressive mechanisms of TKR.

3 A new version of a broadly applicable, cross-lingual meaning representation formalism

3.1 A new version of a mathematical model describing primary units of conceptual level and their interrelations

The first constituent of TKR is a mathematical model (Model 1) describing primary units of conceptual level used by an applied intelligent system and the interrelations of these units.

Let’s use an analogy for explaining the essence of Model 1. It is known that arbitrary language from the class of first order logic languages (or the languages of the first order predicates logic, or FOL languages) is determined by the choice of non-empty set *Const* of symbols called constants, non-empty set *Var* of symbols called variables, non-empty set *F* of functional symbols (the names of functions), non-empty set *Pred* of predicate symbols (the names of *n*-ary predicates, where $n \geq 1$), and a mapping *numb-arg* from the union of *F* and *Pred* associating each symbol from this set with a positive integer *k* interpreted as the number of arguments. That is why it is possible to say that arbitrary language from the class of FOL languages is determined by a complex formal object (a five-tuple) *Logbs* = (*Const, Var, F, Pred, numb-arg*), this formal object may be called *a logical basi*s. Generalizing, we can say that arbitrary language from the class of FOL languages is determined by a five-tuple *Logbs* of the form (*c1, c2, c3, c4, c5*), where *c1, c2, c3, c4, c5* are some formal objects.

Formally, the first version of the Model 1 has the form of the definition of a new class of compound formal objects called *conceptual bases* [8]. According to [8], the construction of an arbitrary conceptual basis (c.b.) *B* is equivalent to selecting a finite sequence of the form *c1, c2, ..., c15*, where *c1, c2, ..., c15* are some formal objects. Each c.b. *B* determines a formal language *Ls(B)* called SK-language (standard knowledge language) in the basis *B*. The expressive mechanisms of SK-languages open broad new prospects for building semantic representations (SRs) of arbitrarily complex sentences and discourses in NL.

The recent version of the Model 1 determines (see the appendix to the present paper) the class of new formal objects called *optimized conceptual bases* (*o.c.b.*). This definition was introduced in [26]. The construction of an arbitrary o.c.b. *Bs* is equivalent to selecting a finite sequence of the form *c1, c2, ..., c9*, where *c1, c2, ..., c9* are some formal objects. Here *c1* = *St* is a finite set of symbols called the sorts and denoting the most general notions from the considered application domains. For instance, *St* may include the elements *phys. ob, org, sit, event* interpreted as the denotations of the notions "physical object", "organization", "situation", "event" (a dynamic situation).

Comparing the number of components 15 of arbitrary c.b. with the number of components 9 of arbitrary o.c.b., it is possible to conclude that, simplifying the Model 1 of TKR, we obtain an optimized version of the theory of SK-languages being the core of TKR. The first basic mathematical model of TKR looks now much simpler, and it will be a great advantage for introducing the university students to TKR.

3.2 A model describing a system of ten partial operations on conceptual structures

The second constituent of TKR is a mathematical model (Model 2) describing a system of such 10 partial operations on structured meanings (SMs) of NL-texts that, using primitive conceptual units as "blocks", we are able to build SMs of arbitrary NL-texts (including articles, textbooks, etc.) and arbitrary pieces of knowledge about the world [8]. These partial operations will be denoted below as Op[1],…, Op[10]. A system consisting of ten partial operations Op[1] – Op[10] is completely mathematically defined in the Chapter 4 of the monograph [8].

Discussing the Model 2, let's mean the updated version of this model. In order to obtain the updated version of the Model 2, it is sufficient to interpret the term "conceptual basis" as the term "optimized conceptual basis" in all definitions and comments of the Chapter 4 of the monograph [8] and to assume that any considered conceptual basis B has the form (*S, Cexp*), where *S* is a sort system, and *Cexp* is an expanded concept-object system coordinated with *S* (see the appendix to the present paper).

Let's consider the algebraic essence of the Model 2. As it is known, a partial algebra on a non-empty set *Y* is any pair *A* of the form (*Y, Z*), where *Z* is a finite set consisting of the partially defined functions from Y^n into *Y*, where $n \geq 1$. The functions from *Z* are called *partial operations on the set Y*. The set *Y* is called *the carrier of the partial algebra A*.

Let *Bs* be an arbitrary optimized conceptual basis, and *L= Ls (Bs)*, $L^1 = L$, $L^2 = L \, x \, L$ (the Cartesian square of the language), for k > 2, $L^k = L \, x \, L \, x \, ...x \, L$ (the Cartesian *k*-degree of *L*), and *Sem-carrier (Ls)* be the union of L^k for all $k \geq 1$. Then each partial operation from the list *Op [1],., Op[10]* is an unary (one-argument) partial operation on the set *Sem-carrier(Ls)*. E.g., let w1 be the logical connective ^ (AND), *w2 = Bratislava, w3 = Ljubljana, w4 = Zagreb*. Then the expression (*Bratislava ^ Ljubljana ^ Zagreb*) is the value of the partial operation *Op [7]* on the four-tuple (*w1, w, w3, w4*). This expression may be a component of an SR of the phrase "George would like to visit this summer Bratislava, Ljubljana, and Zagreb".

Let's see (without numerous mathematical details) how the partial operations Op [1] - Op [10] do work.

The operation Op [1] allows us to join intensional quantifiers and designations (simple or compound) of notions, in particular, for constructing the formulas *certain medicine1, certain medicine1 * (Country-manufacturer, France), certain medicine1 * (Country-manufacturer, (France ∨ Italy)), all medicine1 * (Target-disease, malaria).*

The operation Op[2] is used for constructing the expressions of the form $f(t_1, ..., t_n)$, and Op[3] enables us to build the expressions of the form ($c \equiv d$). Examples: *Height (certain person1), Quantity (all medicine1 * (Target-disease, malaria)) and (Height (certain person1) ≡ 183/cm).*

One uses the operation Op[4] for building the expressions of the form $rel(t_1, ..., t_n)$, where *rel* is the name of a relation with *n* attributes (example: *Belongs (Thomas-Hunt-Morgan, Creators(genetics))).*

The operation Op[5] provides the possibility to mark a formula or its part by means of a variable. Example: *all medicine1 * (Country-manufacturer, India) : S12.*

The operation Op[6] allows us to join the negation connective ¬ to a formula (example: ¬*antibiotic*).

The operation Op[7] governs the usage of the logical connectives ∧ (and) and ∨ (or). Example: *car1 * (Manufacturer, (BMW ∨ Opel)).*

Using the operation Op[8] at the last step of an inference, it is possible to construct compound designations of notions. Example: *medicine1 * (Target-disease, malaria) (Country-manufacturer, India).*

The operation Op[9] allows us to use the universal quantifier and existential quantifier ($\forall$ и $\exists$) in formulas. The operation Op[10] enables us to construct the SRs of finite sequences as the strings of the form < c_1 ,..., c_n >, where c_1 ,..., c_n are the elements of a sequence.

The first and second constituents of TKR form the theory of SK-languages (standard knowledge languages), stated, in particular, in [8]. As a consequence of defining the notion of optimized conceptual basis, the paper [26] introduced a new, more compact version of the theory of SK-languages.

4 From the theory of SK-languages to the theory of K-representations

4.1 Two formal models of a linguistic database

The third constituent of TKR is formed by two broadly applicable mathematical models of a linguistic database (Model 3a and Model 3b). The Model 3a is oriented at Russian language and is introduced in the monograph [27]. The Model 3b is oriented at Russian, English, and German languages and is described in Chapter 7 of the monograph [8]. Both models describe the frames expressing the necessary conditions of the existence of semantic-syntactic relations, in particular, in the word combinations of the following kinds: "Verbal form (verb, participle, gerund) + Preposition + Noun", "Verbal form + Noun", "Noun1 + Preposition + Noun2", "Noun1+ Noun2", "Number designation + Noun", "Attribute + Noun", "Interrogative word + Verb".

4.2 A family of the semantic parsing algorithms

The fourth basic constituent of TKR is formed by a family of complex, strongly structured algorithms carrying out semantic-syntactic analysis (or semantic parsing) of texts from some practically interesting sublanguages of NL. These algorithms transform NL-texts into their semantic representations being K-representations – the SRs being the expressions of SK-languages. The first algorithm called SemSyn is published in Russian in the monograph [27], the input texts of this algorithm form a sublanguage of the Russian language. The next algorithm SemSynt1 is presented in the second part of the monograph [8]. The input texts can be from English, German, and Russian languages. That is why the algorithm SemSynt1 is multilingual.

An important feature of the algorithms SemSyn and SemSynt1 is that they don't construct any syntactic representation of the inputted NL-text but directly find semantic-syntactic relations between text units. The other distinguished feature is that complex algorithms are completely described with the help of formal means, that is why they are problem independent and don't depend on a programming system.

The paper [28] introduces a highly compact way of describing formal structure of linguistic databases (semantic-syntactic component) and of presenting the algorithms of semantic parsing. The paper contains the algorithm of semantic parsing SemSyntRA, developed under the framework of the proposed approach.

4.3 Contributions to several branches of computer science

The fifth constituent of TKR is a collection of scientific results expanding theoretical foundations of advanced ontologies, cross-lingual conceptual information access, Multilingual Semantic Web, the design of agent communication languages in multi-agent systems and recording the content of e-negotiations, semantic parsing of irregular NL-texts: the texts with metonymic phrases and metaphoric expressions [7, 8, 11, 15, 25, 26, 29-31].

5 New prospects for describing semantic content of scientific texts

Let's illustrate only several properties being the principal advantages of TKR by means of constructing semantic representations (SRs) of the definitions D1 – D4 from Section 2.

Property Q1. The possibility to construct SRs of the sets (in essence, it coincides with the property P3 stated in Section 3).

Property Q2. The possibility to construct compound designations of the concepts qualifying the sets (it is a partial case of the property P1).

Example 1. The definition D1 = "The genotype of an organism is the collection of all its chromosomes" from Section 2 (Restriction 2) may have the following SR Semrepr1:

(*Genotype (arbitrary organism: x1) ≡ all gene * (Location, arbitrary chromosome * (Part1, x1))).*

Here the semantic unit *Genotype* is interpreted as the name of a function with one argument.

Example 2. The definition D2 = "Type A blood group are the persons who possess type A isoantigen on red blood cells and anti-B agglutinin in plasma" from Section 2 (Restriction 3) may be associated with the following SR Semrepr2 using a different, general (for the theory of K-representations) structured model:

*Definition (type-A-blood-group, certain set * (Quality-composition, person) : S1, Description(arbitrary person * (Element, S1) : y1, Situation (e1, possessing1 * (Agent1, y1)(Object1, (certain isoantigen * (Type1, 'A')(Location, arbitrary cell1 * (Part1, certain blood1 * (Color, red)(Part1, y1))) ^ certain agglutinin * (Type1, 'anti-B')(Location, certain plasma1 * (Part1, y1))))).*

Property Q3 (in essence, it coincides with the property P4 stated in Section 3). Contrarily to expressive possibilities of first order logic, UCCA, AMR, and UMR, TKR allows us to build formal semantic analogues of the goals, commitments, etc. expressed by infinitive and gerundial constructions.

Property Q4. The possibility to construct SRs of the notions in the form *conc * (rel$_1$, d$_1$) ... (rel$_k$, d$_k$)*, where *conc* is a semantic unit designating a notion, $k≥1$, for $m ≥1$, *rel$_m$* designates either a binary relation or a function with one argument, d_m designates either the second attribute of a binary elation or the value of the function with the name *rel$_m$*.

Property Q5. The possibility to build SRs of the notions' definitions in the form

*(concept1 ≡ concept2 * (rel$_1$, d$_1$) ... (rel$_k$, d$_k$))*, where the unit *concept1* designates the notion to be explained, and *concept2* designates the basic notion used in an explanation of *concept1*.

Example 3. Due to the properties 3 - 5, the definition D3 = "Thrombin is an enzyme which helps to convert fibrinogen to fibrin during coagulation" from Section 2 (Restriction 4) may have the following SR Semrepr3:

(*thrombin ≡ enzyme * (Main-function, helping * (Objective-role, converting1 * (Start-matter, certain fibrinogen) (Final-matter, certain fibrin) (Covering-process, certain coagulation))))*.

Property P6. The possibility to build SRs of complex sentences and discourses with the references to the meanings of phrases and larger parts of discourse.

Example 4. The definition D4 = "All granulocytes are polymorphonuclear; that is, they have multilobed nuclei" from Section 1 (Restriction 5) may have the following SR Semrepr4:

(Property (arbitrary granulocyte: x1, polymorphonuclear): P1 ^ Explanation (P1, Implies (Part1(x1, arbitrary nucleous: x2), Property (x2, multilobed)))).

Property P7. The availability of formal means allowing for constructing object-oriented SRs of definitions, that is, formal structures including such slots as the list of the authors of a definition, the year of its creation, a relevant thematic domain, and SR of the definition.

Example 5. Let D5 = "Control gene is a gene which can turn other genes on or off". Then let

Semrepr5 = (*control-gene ≡ gene * (Is-able, (turning-on * (Object-bio, some gene: Set1) ∧ turning-off * (Object-bio, Set1)))).*

It is possible to construct a different SR of the definition Def5, it will reflect the metadata of information piece, indicating the edition, the authors, and year of publication. In this case

*Semrepr-with-metadata = certain inform-object * (Content1, Semrepr5) (Authorship, (D. Turnpenny ∧ S. Ellard))(Publishing-house, Elsevier)(Year, 2005) (Title, "Emery's Elements of Medical Genetics")(Edition-number, 12).*

The analysis of the scientific literature on artificial intelligence theory, mathematical and computational linguistics shows that today the class of SK-languages opens the broadest prospects for building semantic representations of NL-texts (i.e., for representing structured meanings of NL-texts in a formal way).

SK-languages allow also for describing semantic structure of the sentences with direct and indirect speech and of the discourses with the references to the meanings of phrases and larger parts of a discourse, for constructing compound designations of the notions, sets, and sequences. *As far as one can judge on the available scientific literature, now only TKR explains the regularities of structured meanings of, likely, arbitrary sentences and discourses pertaining to biomedicine and other fields of professional activity of people.*

6 The principal ideas of processing the notions' definitions by the algorithm AlgSemDef1

The input language of the semantic parsing algorithm SemSynt1 introduced in the monograph [8] and mentioned in the subsection 4.2 doesn't include the definitions of the notions. In order to overcome this restriction, the algorithm AlgSemDef1 is presented in [11]. This algorithm uses a little modified form of the algorithm SemSynt1 as a big procedure; the input language of AlgSemDef1 consists of the scholarly notions' definitions having often encountered syntactic structures.

6.1 Input language of an algorithm of definitions' semantic parsing

Let's characterize the input language of the algorithm AlgSemDef1 by means of several examples.

Example 1. Let Def1 = "The Eustachian tube is a canal that leads from the middle ear to the pharynx".

Example 2. Let Def2 = "The Eustachian tube is a canal leading from the middle ear to the pharynx".

Example 3. Let Def3 = "A silent mutation is a mutation not altering the polypeptide product of the gene".

Example 4. Let Def4 = "Pyramid system is the principal efferent pathway of the cortex transmitting the movement impulses, originating in the forward central gyrus of the cortex and reaching the motor neurons of the spinal cord".

Example 5. Let Def5 = "A jack is a technical device for moving physical objects in the vertical plane".

Let's say that the definition Def1 has the type 1, Def2 – Def4 have the type 2, and the definition Def5 has the type 3.

While processing a definition of the type 2, our algorithm initially transforms it into the synonymic definition of the type 1. The meaning of the definition Def5 is the same as of the definition Def 6 = "A jack is a technical device moving physical objects in the vertical plane". That is why, while processing a definition of the type 3, our algorithm will transform it into the synonymic definition of the type 1.

Consider the basic assumptions about the input definitions.

Assumption 1. While elaborating an algorithm of definitions' semantic parsing, we consider not real tokens (nouns, articles, verbs, etc.) but the so called elementary meaningful text units: "a gene", "a technical device", "the piston", "has received", etc.

Example. Consider the definition Def1 = “The Eustachian tube is a canal that leads from the middle ear to the pharynx”. This definition will be regarded as the sequence t_1 t_2 … t_n, where t_1 = “The Eustachian tube”, t_2 = “is”, t_3 = “a canal”, t_4 = “that”, t_5 = leads”, t_6 = “from”, t_7 = “the middle ear”, t_8 = “to”, t_9 = “the pharynx”, t_{10} = dot.

Assumption 2. Let’s suppose that all input definitions have the form _t1 t2 t3 descr_, where t_1_ is a lexical representation (LR) of the notion to be explained, _t2_ is the word “is”, t_3_ is a LR of the basic notion, _descr_ is a fragment describing the properties possessed by the objects qualified by the notion to be explained.

6.2 The central ideas of definitions’ semantic parsing

The monograph [8] introduces an original, strongly structured algorithm of semantic parsing (in other terms, an algorithm of semantic-syntactic analysis) called the algorithm SemSynt1. This algorithm is multilingual: the input texts may belong to restricted sublanguages of English, German, and Russian (in the last case the texts in Latin transcription are considered). The input texts may be the questions of many kinds, the statements, and the commands. The algorithm SemSynt1 transforms an input text into its semantic representation being its K-representation (knowledge representation), i.e., an expression of a certain SK-language.

The algorithm of semantic parsing SemSynt1 [8] provided by TKR opens broad prospects of developing useful for practice algorithms for semantic processing of knowledge pieces expressed in NL, in particular, of the definitions of the notions.

Example. Let’s consider the central ideas of processing the definition Def4 = “Pyramid system is the principal efferent pathway of the cortex transmitting the movement impulses, originating in the forward central gyrus of the cortex and reaching the motor neurons of the spinal cord” by the algorithm AlgSemDef1 introduced in [11].

Let’s assume that the knowledge base includes the units _pyramid-system_ and _pathway1_. The definition Def4 contains the verb produced forms “transmitting”, “originating”, and “reaching”. That is why the variable _nvb_ (the number of verb produced forms) receives the value 3. The execution of the algorithm AlgSemDef1 includes the construction and semantic parsing of the auxiliary sentences Sent1 = “The principal efferent pathway of the cortex transmits the movement impulses”, Sent2 = “The principal efferent pathway of the cortex originates in the forward central gyrus of the cortex”, and Sent3 = “The principal efferent pathway of the cortex reaches the motor neurons of the spinal cord”.

In the cycle for _i_ from 1 to _nvb_, the variable _phrase_ receives the value Sent1 in case i = 1, the value Sent2 in case i = 2, and the value Sent3 in case i = 3.

The one-dimensional array Sem1 with the elements being strings is used for storing the primary semantic representations (SRs) of the sentences Sent1, Sent2, and Sent3. These primary SRs are constructed by the algorithm SemSynt1 (in a little modified form). The following assignments will be fulfilled (in the context of a certain linguistic database) in the cycle for _i_ from 1 to _nvb_ (the element _certn_ is an intensional quantifier corresponding to the meaning of the word combination “a certain”, the element _Qual-compos_ denotes the binary relation “Qualitative composition of a set”):

Sem1[1] := _Situation(e1, trasmission1 * (Mediator1, certn pathway1 * (Importance1, principal)(Quality1, efferent)(Part, certn cortex1 : x2) : x1)(Object1, certn set * (Qual-compos, impulse1 * (Purpose1, movement)) : S3)),_

Sem1[2] := _Situation(e1, originating * (Focus-entity, certn pathway1 * (Importance1, principal)(Quality1, efferent)(Part, certn cortex1 : x2) : x1)(Departure-entity, certn gyrus1 * (Space-property1, forward)(Space-property2, central)(Part, certn cortex1 : x4) : x3)),_

Sem1[3] := _Situation(e1, reaching1 * (Focus-entity, certn pathway1 * (Importance1, principal)(Quality1, efferent)(Part, certn cortex1 : x2) : x1)(Destination-entity, certn set * (Qual-compos, neuron1 * (Purpose1, setting-in-motion)(Part, certn spinal-cord : x4) : S3))._

The purpose of the next stage of executing the algorithm AlgSemDef1 is to improve the referents of various entities mentioned in the sentences Sent1, Sent2, and Sent3. Firstly, it is easy to see that the variables x2 in the expression Sem1[1] and the variable x4 in the expression Sem1[2] are the marks of the same cortex. That is why the variable x4 is to be replaced by the variable x2 in the expression Sem1[2]. Secondly, the variable S3 denotes a set of movement impulses in Sem1[1] and a set of motor neurons in Sem1[3]. That is why the variable S3 is to be replaced by the variable S4 in the expression Sem1[3].

Taking into account these considerations, let’s fulfill a number of actions. Let _Concrepr2_ be an expression obtained from the expression _Sem1[2]_ by means of replacing the variable _e1_ by _e2_ and the variable _x4_ by the variable _x2._ Let _Concrepr3_ be an expression obtained from the expression _Sem1[3]_ by means of replacing the variable _e1_ by _e3_ and the variable _S3_ by the variable _S4._ Then let

Sem2[1] := Sem1[1], Sem2[2] := Concrepr2, Sem2[3] := Concrepr3.

Thus, the array Sem2 will have the configuration reflected on Table 2.

Table 2: The configuration of the array Sem2 constructed from the input definition Def4 by the algorithm AlgSemDef1.

Situation (e1, trasmission1 * (Mediator1, certn pathway1 * (Importance1, principal) (Quality1, efferent) (Part, certn cortex1 : x2) : x1)(Object1, certn set * (Qual-compos, impulse1 * (Purpose1, movement)) : S3))
Situation (e2, originating * (Focus-entity, certn pathway1 * (Importance1, principal) (Quality1, efferent) (Part, certn cortex1 : x2) : x1)(Departure-entity, certn gyrus1 * (Space-property1, forward)(

Space-property2, central)(Part, certn cortex1 : x2) : x3))
*Situation (e3, reaching1 * (Focus-entity, certn pathway1 * (Importance1, principal) (Quality1, efferent) (Part, certn cortex1 : x2) : x1)(Destination-entity, certn set * (Qual-compos, neuron1 * (Purpose1, setting-in-motion)(Part, certn spinal-cord : x4) : S4))*

Then the algorithm AlgSemDef1 builds a K-representation of Def4 as the formula *Sem-final* of the form *Definition* (*pyramid-system, pathway1, x1,* (*Sem2[1]* $\wedge$ *Sem2[2]* $\wedge$ *Sem2[3]*)).

This formula is interpreted as follows: an arbitrary *pyramid system* is denoted by the variable *x1*, the physical object x1 is a pathway1 (a pathway in a biological object), and the formula (*Sem2[1]* $\wedge$ *Sem2[2]* $\wedge$ *Sem2[3]*) describes the properties of the physical object *x1*.

7 Improvement of the algorithm AlgSemDef1

Analysing the structure of the one-dimensional array Sem2 constructed by the algorithm AlgSemDef1 from the input definition Def4 (see Table 2), we see that each of the elements Sem2[1], Sem2[2], Sem2[3] includes the same fragment being a semantic image of the word combination “the principal efferent pathway of the cortex”. It is the expression

*certn pathway1 * (Importance1, principal) (Quality1, efferent) (Part, certn cortex1 : x2) : x1.* (1)

Let’s develop a simple algorithm Improve-actants replacing by x1 all occurences of a value associated with the first realized semantic role in the expressions Sem2[m], where m>1. In case of processing the definition Def4 (it explicates the notion “a pyramid system”), this algorithm will replace the second and third occurences of the expression (1) by the variable x1.

Let’s agree that the elements of the considered informational universe X(B) and the variables from the set V(B), where B is a conceptual basis, will be interpreted below as symbols.

Algorithm Improve-actants

External specification

Input: one-dimensional array Sem2 containing the strings.
Output: transformed one-dimensional array Sem2.

Algorithm

```
Begin
Cycle for k from 2 to nvb
Begin
Let p be the position of the first occurrence (from the left) of the variable x1 in Sem2[k]. Then delete the symbols in the positions 10, 11, p-1 of the expression Sem2[k]
End
```

Example. Let Sem2 be the array constructed by the algorithm AlgSemDef1 for the input Def4 (see Fig. 1). Then the correspondence between the positions 1, 2, …, 12 and the symbols in these positions forming the string Sem2[2] is shown on Table 3.

Table 3: A correspondence between the positions of a left fragment of Sem2[2] and the elements (the symbols) in these positions.

position	element
1	Situation
2	(
3	e2
4	,
5	originating
6	*
7	(
8	Focus-entity
9	,
10	certn
11	pathway1
12	*

Then the first three symbols deleted by the algorithm *Improve-actants* will be the elements *certn, pathway1, *.* Let p be the position of the first occurrence of the variable x1 in the expression Sem2[2]. Then the last deleted symbol will be the colon in the position p-1 (the left neighbour of the element x1 in the expression Sem2[2]). The algorithm AlgSemDef1 is the sequence of the auxiliary algorithms FirstStage, Rename-variables, and FinalStep. *Then let’s define the algorithm AlgSemDef2 as the sequence of the auxiliary algorithms FirstStage, Rename-variables, FinalStep, and Improve-actants.*

Example. Processing the input definition Def4 (an explication of the notion “a pyramid system”), the algorithm AlgSemDef2 will construct the array Sem2 whose configuration is reflected on Table 4.

Table 4: The configuration of the array Sem2 constructed from the input definition Def4 by the algorithm AlgSemDef2.

*Situation (e1, trasmission1 * (Mediator1, certn pathway1 * (Importance1, principal) (Quality1, efferent) (Part, certn cortex1 : x2) : x1)(Object1, certn set * (Qual-compos, impulse1 * (Purpose1, movement)) : S3))*
*Situation (e2, originating * (Focus-entity, x1) (Departure-entity, certn gyrus1 * (Space-property1, forward) (Space-property2, central) (Part, certn cortex1: x2) : x3))*
*Situation (e3, reaching1 * (Focus-entity, x1) (Destination-entity, certn set * (Qual-compos, neuron1 * (Purpose1, setting-in-motion) (Part, certn spinal-cord: x4) : S4))*

Then the algorithm AlgSemDef2 builds a K-representation of Def4 as the formula *Sem-final* of the form *Definition* (*pyramid-system, pathway1, x1,* (*Sem2[1]* $\wedge$ *Sem2[2]* $\wedge$ *Sem2[3]*)), that is, of the form

Definition (*pyramid-system, pathway1, x1,* (*Situation(e1, trasmission1 * (Mediator1, certn pathway1 * (Importance1, principal)(Quality1, efferent)(Part, certn cortex1 : x2) : x1)(Object1, certn set * (Qual-compos, impulse1 * (Purpose1, movement)) : S3))* ∧ *Situation(e2, originating * (Focus-entity, x1)(Departure-entity, certn gyrus1 * (Space-property1, forward)(Space-property2, central)(Part, certn cortex1 : x2) : x3))* ∧ *Situation(e3, reaching1 * (Focus-entity, x1)(Destination-entity, certn set * (Qual-compos, neuron1 * (Purpose1, setting-in-motion)(Part, certn spinal-cord : x4) : S4)))).*

8 Experimental validation of the algorithm AlgSemDef2

The suggested algorithm AlgSemDef2 is implemented in the computer program developed under the framework of the master thesis [32]. The program is realized by means of the language Java and uses the system PostgreSQL as a database management system. The program executes a two-step transformation of a knowledge piece description in Russian. The first step is fulfilled by the algorithm 1. Its content is the transformation "NL description of a knowledge piece => K-representation". The second step (fulfilled by the algorithm 2) is the transformation "K-representation of a knowledge piece => A collection of the records in the language OWL (Ontology Web Language)", the goal is to inscribe the created collection of the OWL-records into an OWL-based ontology.

The algorithm 1 includes the algorithm AlgSemDef2 as a complex procedure, but the input language is broader than in case of the algorithm AlgSemDef2. Any input of the algorithm 1 has the form *Def Addition 1…. Addition N,* where $N \geq 1$, the fragment *D*ef is a definition of a notion *conc*, $N \geq 1$, the fragment *Addition 1…. Addition N* states additional information about the objects qualified by the notion *conc*. In particular, the input language of the algorithm 1 includes the knowledge piece "Crocodile is an animal from the order of water reptiles. The crocodiles live in the rivers, lakes, and marches. Geographical zone is tropics. The length may reach 7 m".

9 Conclusion

It is shown above that the theory of K-representations (TKR) provides a number of precious expressive mechanisms for describing in a formal way semantic structure of scholarly sentences and discourses in NL pertaining to biology and medicine. With this aim, TKR introduced (in several previous publications) a new class of formal languages – the class of SK-languages (standard knowledge languages). TKR suggests to use this class of languages for building semantic representations (SRs) both of sentences and discourses. The main attention is paid to the advantages of TKR in comparison with Universal Conceptual Cognitive Annotation, Abstract Meaning Representation, and Uniform Meaning Representation.

The are serious reasons to conjecture that the considered optimized version of the SK-languages' theory (a part of TKR) may be interpreted as the *starting universal meaning representation* (*MR*) *formalism*. In this connection, the new, very topical task for the computer linguists of many countries is to select the collections of semantic units being convenient for forming MRs of sentences and discourses in their native languages, using ten rules of the SK-languages' theory for combining these semantic units into (a) MRs of sentences and discourses, (b) conceptual structures emerged in the memory of applied intelligent systems (AIS) during a dialogue with a human being or another AIS.

The topicality of the problem of forming and updating ontologies by means of automatic extraction of knowledge from NL-texts motivated the development of an algorithm of building SRs of the scholarly notions' definitions (it was published in a previous work of the present paper's author and is called AlgSemDef1). TKR underpinned the creation of the algorithm AlgSemDef1. The present paper improves the algorithm AlgSemDef1 and defines the algorithm AlgSemDef2. The purpose is to make more compact the output of the algorithm. The main steps of processing complex scholarly definitions pertaining to biology and medicine by the algorithms AlgSemDef1 and AlgSemDef2 are illustrated.

Thus, it is shown above that TKR opens new precious prospects for the designers of NL processing systems in biomedical sciences.

Acknowledgement

I am grateful to the anonymous referees for precious comments helped to improve this paper.

Appendix: The definition of optimized conceptual basis

The definitions considered below were initially introduced in [26].

A. Sort systems

Let's imagine that we should start the formalization of a certain application domain or a group of domains. Our aim will be the ability to formally represent semantic structure of arbitrary NL-texts used in the considered domain. At the first step, let's select a finite set *St* of sorts and distinguish in *St* a certain sort *P* to be called the sort "a meaning of proposition" (in the sense "a meaning of statement"). This sort *P* will be interpreted as the mark (the type) of semantic representations (SRs) of sentences being statements and of arbitrary narrative discourses.

Let's suppose that the content of next two steps is to define two binary relations *Gen* and *Tol* on the set *St.* It means that we define two subsets of the Cartesian product *St* × *St.* The relation *Gen* formalizes the hierarchy on the set of most general notions *St.* Our requirement will be that it is a reflexive, antisymmetric, and transitive relation (i.e., a partial order on *St*). Due to reflexivity, for arbitrary sort w, the pair (*w, w*) belongs to *Gen.* The pair (*real, integer*) belongs to *Gen* means that all integers form a subclass of all real numbers.

The binary relation *Tol* is to formalize the phenomenon of the existence of two or more different angles of look at some objects. For instance, a scholar with usual physical capabilities may be considered, on the one hand, as an intelligent system and, on the other hand, as a dynamic physical object (because she/he may run, spring, etc.). Taking this into account, the pair (*ints, dyn. phys. ob*) belongs to *Tol* in case the sorts *ints* and *dyn. phys. ob* are interpreted respectively as "an intelligent system" and "a dynamic physical object".

Definition 1 (introduced in [6]). A sort system is an arbitrary four-tuple *S* of the form (*St, P, Gen, Tol*), where *S* is an arbitrary finite set of symbols, *P* belongs to *St*, *Gen* is a non-empty binary relation on *St* being a partial order on *St* (a reflexive, antisymmetric, and transitive relation), *Tol* is a binary relation on *St* being antireflexive and symmetric, and the following conditions are satisfied:

(1) *St* doesn't include the symbols ↑, *, {,}, (,), [*entity*], [*concept*], [*object*], [↑*entity*], [↑*concept],* [↑*object*].

(2) If *Concretizations(P)* is the set of all such *z* from the set *St* that (*P, z*) belongs to *Gen* then the set-theoretical difference *St \ Concretizations(P)* is not empty, and for every *u* from *St \ Concretizations(P)* and for every *w* from *Concretizations(P)* the sorts *u* and *w* are incomparable both for the relation *Gen* and for the relation *Tol*.

(3) For each *t, u* from *St*, it follows from (*t, u*) belongs to *Gen* or (*u, t*) belongs to *Gen* that *t* and *u* are incomparable for the relation *Tol*.

(4) For each *t1, u1* from *St* and each *t2, u2* from *St*, it follows from (*t1, u1*) belongs to *Tol*, (*t2, t1*) belongs to *Gen*, (*u2, u1*) belongs to *Gen* that (*t2, u2*) belong to *Tol*.

The elements of the set *St* are called sorts, *P* is called the sort "a meaning of proposition", the binary relations *Gen* and *Tol* on the set *St* are called respectively the *generality relation* and *tolerance relation*. If *t, u* belong to *St* and the pair (*t, u*) belongs to *Gen*, then we often use an equivalent notation *t* →*u* and say that *t* is a generalization of *u*, and *u* is a concretization of *t.*

The symbols listed in the requirement (1) play special roles in TKR. The requirement (2) is to be interpreted as follows. The sort *P* will play the role of the type (the mark) of semantic representations (SRs) of statements and narrative texts. Suppose that we have a pragmatic reason to introduce the particular cases of the sort *P:* the sort *P-act* as the type of the statements about the physical actions, *P-inf* as the type of the statements about information transmissions, and *P-comm* as the type of the statements being implicit commands ("It is necessary to close the dor", etc.). Then *Concretizations (P)* = {*P, P-act, P-inf, P-comm*}.

But we should have different sorts (let they form a subset *Usual-things-sorts* of the set *St*) for classifying things to be mentioned in the statements of the mentioned kinds. Obviously, for every *u* from the subset *Usual-things-sorts* and for every *w* from *Concretizations(P)* the sorts *u* and w should be incomparable both for the relation *Gen* and for the relation *Tol.*

B. The set of types corresponding to a sort system

The monograph [8] contains a definition associating arbitrary sort system *S* with the countable set *Tp(S)* of strings called types and interpreted as simple and compound semantic characteristics of the entities from the considered application domains.

Example. There is such sort system S_1 that the set $Tp(S_1)$ includes, in particular, the elements *ints* * *phys.ob*, {*ints* * *phys.ob*}, ↑*ints* * *phys.ob, integer*, {*integer*}, (integer, integer), {(*integer, integer*)}. The interpretation of these types will be considered below in an example immediately after the definition 2.

The arrow ↑ distinguishes the types of the notions; the types of the form {*z*} are the types of the sets; the types of the forms (*x, y*), (*x, y, z*) are respectively the types of a pair and of a triple, etc. The type {(*integer, integer*)}is to be interpreted as the type of various binary relations on the set of all integers; in particular, as the type of the relations < and >. The form of the type *ints* * *phys.ob* says that the pair (*ints, phys.ob*) belongs to *Tol.*

C. Expanded concept-object systems

Definition 2. Let *S* be any sort system of the form (*St, P, Gen, Tol*). Then a five-tuple *Cexp* of the form (*X, V, tp, F, ref*) is called an *expanded concept-object system* (*e.c.o.s.*) *coordinated with the sort system S* then and only then the following conditions are satisfied:

X and *V* are countable non-intersecting sets of symbols; *tp* is a mapping from the union of *X* and *V* into the set of types *Tp(S)*;

F is a subset of *X*; for each *h* from *F*, the string *tp(h)* has the beginning {(and the ending)};

The set of sorts *St* is a subset of *X*, and for each *w* from *St, tp(w) =* ↑*w; ref* is a distinguished element of the set-theoretical difference *X \ Y*, where *Y is* the union of *St* and *F*;

The set of all such elements *var* from *V* that *tp(var)* = [*entity*] is countable.

The set *X* is called the *primary informational universe*; the elements of *V* are called *variables*. The elements of the subset *F* are called *functional symbols* (or the names of functions). The distinguished element *ref* is called the *referential quantifier*.

The referential quantifier *ref* is interpreted as the semantic unit designating the meaning of the expression "a certain". In the examples in English usually *ref* is the string *certain* or *certn*.

The mapping *tp* gives us a much more fine-grained structuring of application domains than first order logic.

Example. The sets *St*, *X* and the mapping *tp* may satisfy the following conditions: (a) *St* includes the elements (sorts) *dyn.phys.ob* (dynamic physical object), *ints* (intelligent system), *org* (organization), *inf.ob* (informational object); (a) *X i*ncludes the elements

L.Carroll, *Alice-in-Wonderland*, *person*, *student-group*, *Suppliers*, *Authorship,* and

tp(person) =↑ *ints * dyn.phys.ob, tp(L.Carroll) = ints * dyn.phys.ob; tp(Alice-in-Wonderland) = inf.ob, tp(Authorship) = {(ints, inf.ob)};*

*tp(student-group) = =↑ {ints * dyn.phys.ob}, tp(Suppliers) = {(org, {org})}.*

Here the symbol ↑indicates the type of a notion; *Suppliers* is the name of the function associating an enterprise with the set of all its suppliers.

D. Optimized conceptual bases

Definition 3. Let *S* be any sort system of the form (*St*, *P*, *Gen*, *Tol*), and a five-tuple *Cexp* of the form (*X*, *V*, *tp*, *F*, *ref*) be an expanded concept-object system coordinated with the sort system *S*. Then the ordered pair *B* = (*S*, *Cexp*) will be called an optimized conceptual basis if and only if the following conditions are satisfied:

The sets *X* and *V* don't include the symbols ',' (comma), '*', ':', '(', ')', '<', '>', '&';

The set of sorts *St* includes the symbols *eqv*, *neg*, *binlog*, *int1*, *int2*, *ext*; the primary informational universe *X* includes the universal and existential quantifiers and, besides, the symbols ≡, ¬ , ∨ , ∧;

The union of the set *Int1={y from X| tp(y) = int1}* and the set *Int2={w from X| tp(w) = int21}* belongs to the set-theoretical difference *X \ Y*, where *Y* is the union of *St* and *F*; *tp(ref) = int1*;

tp(≡) = *eqv*; tp(¬) = *neg*; tp(∨) = tp(∧) = *binlog*; the value of the mapping *tp* for the universal quantifier and existential quantifier is equal to ext.

The elements of the sets *Int1* and *Int2* will be called *intensional quantifiers.*

The symbols ≡, ¬ , ∨ , ∧ should be read as "identical to", "not", "or", "and". The elements of the set *Int1* are interpreted as semantic units denoting the meanings of the words and expressions "a certain", "any", "arbitrary", etc. The elements of the set *Int2* are interpreted as semantic units denoting the meanings of the words and expressions "all", "several", "many", "a few", etc.

Using these denotations and interpreting in Chapter 4 of [8] the symbol *B* as an optimized conceptual basis (but not as a conceptual basis), we are able to get the definition of the SK-language *Ls(B)* in the basis *B*. Thus, due to the simplification of the basic Model 1 of TKR, we obtain a simplified (or optimized) version of the theory of SK-languages.

References

[1] Van Gysel, J. E. L., Vigus, M. et al. (2021). Designing a Uniform Meaning Representation for Natural Language Processing. *Kuenstliche Intelligenz*, vol. 35, no. 3-4, pp. 1-18. https://doi.org/10.1007/s13218-021-00722-w

[2] Bonn, J., Flanigan, J., Hajic, J., Jindal, I., Li, Y., and Xue, N. (2024). Meaning Representations for Natural Languages: Design, Models and Application. In *Proceedings of the 2024 Joint Intern. Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 20-25 May 2024,* Language Resources Association (ELRA), 2024, pp. 13-18; https://aclanthology.org/2024.lrec-tutorials.3.pdf

[3] DMR-1 Workshop (2019). *Proceedings of DMR 2019: The First International Workshop on Designing Meaning Representation*. Italy, Florence, August 1, 2019 in conjunction with ACL 2019; https://aclanthology.org/W19-33.pdf

[4] DMR-5 Workshop (2024). *Proceedings of DMR 2024: The Fifth International Workshop on Designing Meaning Representation*, Italy, Torino, May 21, 2024 in conjunction with 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Language Resources Association (ELRA), 2024; https://aclanthology.org/2024.dmr-1.0

[5] Zhu, H., Li, Y. and Chiticariu, L. (2019). Towards universal semantic representation. In *Proceedings of DMR 2019: The First International Workshop on Designing Meaning Representation, Italy, Florence, August 2019 in conjunction with ACL 2019,* pp. 177-181. https://doi.org/10.18653/v1/w19-3320

[6] Fomichov, V.A. (1996). A Mathematical Model for Describing Structured Items of Conceptual Level. *Informatica. An International Journal. of Computing and Informatics (Slovenia)*, vol. 20 no. 1, pp.15-32.

[7] Fomichov, V.A. (2008). A Comprehensive Mathematical Framework for Bridging a Gap between Two Approaches to Creating a Meaning-Understanding Web. *Intern. Journal of Intelligent Computing and Cybernetics*, vol. 1 no. 1, pp. 143-163. https://doi.org/10.1108/17563780810857176

[8] Fomichov, V.A. (2010a). *Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms.* Springer, New York, Dordrecht, Heidelberg, London, 352 p. https://doi.org/10.1007/978-0-387-72926-8

[9] Abend, O. and Rappoport, A, (2013). Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics,* Sofia, Bulgaria, August 4-9 2013. Association for Computational Linguistics, 2013, pp. 228-238; https://aclanthology.org/P13-1023

[10] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M. and Schneider, N. (2013). Abstract Meaning Representation for Sembanking. *Proceedings of the 7th ACL Linguistic Annotation*

Workshop and Interoperability with Discourse, Sofia, Bulgaria, August 8-9, 2013 (www.aclweb.org/anthology/W13-2322; retrieved 2024-07-30)

[11] Fomichov, V.A. (2022). Semantic Mapping of Definitions for Constructing Ontologies of Business Processes. In *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), May 23 - 27, 2022, Opatija, Croatia. Proceedings*. Croatian Society for Information, Communication and Electronic Technology - MIPRO, Rijeka, 2022, pp. 1258-1263. https://doi.org/10.23919/mipro55190.2022.9803553

[12] Auer, S., Bryl, V. and Tramp, S. (Editors) (2014). *Linked Open Data -- Creating Knowledge Out of Interlinked Data. Results of the LOD2 Project*, Springer, 2014; open access at https://link.springer.com/book/10.1007/978-3-319-09846-3. https://doi.org/10.1007/978-3-319-09846-3

[13] Blaney, J. (2017). Introduction to the Principles of Linked Open Data. *Programming Historian*, 2017; open access at https://programminghistorian.org/en/lessons/intro-to-linked-data. https://doi.org/10.46430/phen0068

[14] Ernst, P., Siu, A., and Weikum, G. (2015). KnowLife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC Bioinformatics*, 16, 157 (2015); open access at https://doi.org/10.1186/s12859-015-0549-5 (retrieved 2024-07-30).

[15] Fomichov, V. A. (2011). The prospects revealed by the theory of K-representations for bioinformatics and Semantic Web. *Actes de la 18e conference sur le Traitement Automatique des Langues Naturels. Actes de la 15e Rencontre des Etudiants Cercheurs en Informatique pour le Traitement Automatique des Langues. France, Montpellie, 27th June - 1st July 2011 Vol. 1: Actes: articles longs*, AVL Diffusion, Montpellier, pp 5-20; https://aclanthology.org/2011.jeptalnrecital-invite.1.pdf

[16] Wang, C., Xue, N., and Pradnan, S. (2015). A transition-based algorithm for AMR parsing. In *Proc. of the 2015 Conf. of the North American Chapter of ACL: Human Language Technologies. Denver, Colorado*, ACL, 2015, pp. 366-375. https://doi.org/10.3115/v1/n15-1040

[17] Flanigan, J., Thomson, S., Carbonell, J., Dyer, C., and Smith, N.A. (2014). A discriminative graph-based parser for the abstract meaning representation”. In *Proc. of the 52nd Annual Meeting of ACL, Vol. 1: Long Papers*, Baltimore, Maryland, ACL, 2017, pp. 1426-1436. https://doi.org/10.3115/v1/p14-1134

[18] Pust, M., Hermjakob, U., Knight, K. , Marcu, D. and May, J. (2015). Parsing English into abstract meaning representation using syntax-based machine translation. In *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing. Lisbon, Portugal*, ACL, 2015, pp. 1143-1154. https://doi.org/10.18653/v1/d15-1136

[19] May, J. (2016). SemEval-2016 task 8: Meaning representation parsing, In *Proc. of the 10th Intern. Workshop on Semantic Evaluation (SemEval-2016).* ACL, San Diego, California, 2016, pp. 1063-1073. https://doi.org/10.18653/v1/s16-1166

[20] Damonte, M., Cohen, S.B., and Satta, G. (2017). An incremental parser for abstract meaning representation. *Proc. of the 15th Conference of the European Chapter of ACL, April 2017, vol. 1, Long Papers*, Valencia, Spain, ACL, 2017, pp. 536-546. https://doi.org/10.18653/v1/e17-1051

[21] Rao, S., Marcu. D. and Hal Daume III. (2017). Biomedical event extraction using Abstract Meaning Representation. In *Proc. of the Intern. Conference BioNLP 2017, Vancouver, Canada, Aug. 2017.* ACL, 2017, pp. 126-135. https://doi.org/10.18653/v1/w17-2315

[22] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M. and Schneider, N. (2019). *Abstract Meaning Representation (AMR) 1.2.6 Specification. May 1, 2019*; github.com/amrisi/amr-guidelines/blob/master/amr.md (retrieved 2024-07-30).

[23] Zhao, J., Xue, N., Van Gysel, J., and Choi, J. D. (2021). UMR-Writer: A Web Application for Annotating Uniform Meaning Representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Nov. 7-11, 2021.* Association for Computational Linguistics, 2021, pp. 160-167. https://doi.org/10.18653/v1/2021.emnlp-demo.19

[24] Bonn, J., Buchholz, M., et al. (2024). Building a Broad Infrastructure for Uniform Meaning Representation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 20-25 May 2024*. Language Resources Association (ELRA), 2024, pp. 2537-2547; https://aclanthology.org/2024.lrec-main.229.pdf

[25] Fomichov, V.A. (2017). SK-languages as a Powerful and Flexible Semantic Formalism for the Systems of Cross-Lingual Intelligent Information Access. *Informatica. An Intern. J. of Computing and Informatics (Slovenia)*, 2017, vol. 41, pp. 221-232.

[26] Fomichov, V.A. (2021). Intelligent Monitoring of News on Economics and Finance Based on Formal Semantics of the Movement Verbs. In *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), September 27 – October 1, 2021, Opatija, Croatia. Proceedings*, Croatian Society for Information, Communication and Electronic Technology - MIPRO, Rijeka, 2021, pp. 1253-1258. https://doi.org/10.23919/mipro52101.2021.9597096

[27] Fomichov, V.A. (2005). *The Formalization of Designing Natural Language Processing Systems*, Moscow, MAKS Press, 368 p. (in Russian).

[28] Razorenov, A. A. and Fomichov, V. A. (2016). A new formal approach to semantic parsing of instructions and to file manager design Database and Expert Systems Applications. In *Proceedings of the 27th International Conference, DEXA 2016, Porto, FomichovInformaticaCameraReadyPaper2024fin1.dotxPortugal, September 5-8, 2016,* Springer, Cham, 2016, vol. 9827. Part. I, pp. 416-430. https://doi.org/10.1007/978-3-319-44403-1_26

[29] Fomichov, V.A. (2010b). Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web. *Informatica. An Intern. Journal of Computing and Informatics*, 2010, vol. 34, pp. 387-396.

[30] Fomichov, V.A. (2013). A Broadly Applicable and Flexible Conceptual Metagrammar as a Basic Tool for Developing a Multilingual Semantic Web. In *Metais, E., Meziane, F., Saraee, M., Sugumaran, V., Vadera, S. (eds.) NLDB 2013. LNCS, 2013,* vol. 7934. Springer, Heidelberg. 2013, pp. 249-259. https://doi.org/10.1007/978-3-642-38824-8_21

[31] Fomichov, V.A. (2023). A New Approach to Semantic Parsing of Metonymic Phrases by a Business Intelligence System. In *2023 46th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 22 - 26, 2023, Opatija, Croatia. Proceedings*, Croatian Society for Information, Communication and Electronic Technology - MIPRO, Rijeka, 2023, pp. 1305-1310. https://doi.org/10.23919/mipro57284.2023.10159720

[32] Urubkov, V. S. (2024). Development of a Semantic Analyzer of Natural Language Knowledge Descriptions. Master thesis (Academic Advisor Prof. V. A. Fomichov). Moscow Aviation Institute (National Research University), 2024.

A Proposed Paradigm Using Data Mining to Minimize Online Money Laundering

Shimaa Ouf [1*], Meram Ashraf [2], Mohamed Roushdy [3]
[1,2] Information Systems Department, Faculty of Commerce and Business Administration, Helwan University, Helwan, Egypt
[3] Faculty of Computers & Information Technology, Future University in Egypt New Cairo, Cairo, Egypt
E-mail: shimaaouf@commerce.helwan.edu.eg, Meram.ashraf21@commerce.helwan.edu.eg, Mohamed.Roushdy@fue.edu.eg
[*]Corresponding author

Keywords*:* online money laundering, data mining, clustering, classification, association

Received: April 21, 2024

Since the global financial crisis (GFC), banks have been compromised by various risks. One of the significant risks is online money laundering. It is the third-largest business in the world after currency exchange and the automotive industry. As technology has advanced, the methods of online money laundering have become more evasive. Banks' traditional methods cannot deal with online money laundering. The absence of contemporary anti-money laundering techniques has led to the rise of this criminal activity. As a result, the existing systems need effective technology to accommodate the development of online money laundering. Data mining is the technology that applies mathematical, statistical, and machine-learning techniques to extract patterns like malicious behavior of money launderers and gain information about whether the online transaction belongs to money laundering. Many research papers apply different data mining techniques to predict online money laundering. By analyzing these research papers, we found that authors focused on applying one or two data mining techniques to predict online money laundering but ignored combining many techniques like classification, clustering, and association to improve the accuracy of prediction and overcome the limitations of each of them. Therefore, this paper proposes and implements a paradigm (APPD-OML) based on classification, clustering, and association techniques to improve the accuracy of predicting and detecting online money laundering. The result of testing the proposed paradigm illustrates that the prediction and detection of online money laundering based on applying data mining techniques like classification, clustering, and association achieve a strong accuracy of 94%, f-measure of 95%, and AUC 95% which means that the proposed paradigm outperforms each technique used separately in predicting and detecting online money laundering and outperformed the other research that used data mining in this field.

Povzetek: V članku je predstavljena nova paradigma APPD-OML, ki združuje tehnike klasifikacije, združevanja in asociacije za izboljšanje napovedovanja in zaznavanja spletnega pranja denarja..

1 Introduction

The advancement of technology, along with the COVID-19 pandemic, has prompted businesses and the public to use digital technologies even more intensively. Online services, online payments, and the use of electronic payment systems have significantly increased [1].

Despite the many advantages of online transactions, including time savings, reduced sales queues, a decrease in the need for cash and checks, ease of management, 24-hour accessibility, etc. Online transactions remain a double-edged weapon because they are a fertile environment for malicious attackers and criminals to do what they want without any tracking or monitoring [2].

Many criminals use online transactions as a way of laundering their money [3]. Money laundering is the process of giving fraudulent (dirty) money a legitimate look. It is the process of changing or converting assets that came from a criminal source to assist the criminal who is conducting the crime [4].

Money laundering has a destructive economic effect and relationship to the financing of terrorism. It weakens the stability and integrity of financial institutions as well as national economic stability in a way that distorts global capital flows and deters foreign direct investment [4]. Due to the digital era, many criminals use the internet to turn monetary gains from illegal activities into "clean" funds, which is known as online money laundering [3].

Online money laundering is a sophisticated practice that is carried out via a variety of cyber media, such as e-commerce, e-banking, online gaming, online gambling, electronic money, and other cyber means [5].

Therefore, banks' existing approaches are insufficient for predicting and detecting such activity. As a result, the requirement for automated prediction and detection technology, such as data mining, became essential [3].

Data mining is a method for locating hidden information in datasets. It is the process of extracting and identifying potential and valuable knowledge stored in huge amounts of data by utilizing statistical techniques, mathematics, artificial intelligence, and machine learning [6]. In recent years, researchers have focused a lot of attention on the role that data mining plays in predicting and detecting money laundering in general and online money laundering in particular. Many methods are investigated, proposed, and studied for providing anti-money laundering solutions [7]. For such reasons, the main goal of this paper is to present a paradigm based on data mining techniques to predict and detect online money laundering.

The rest of this paper is divided as follows: Section 2 presents the research methodology and related work related to the topic. In Section 3, the proposed paradigm is presented. Section 4 demonstrates the experiments and results. In Section 5, the evaluation is discussed. In Section 6, the discussion is presented and finally, Section 7 presents the conclusion and future research.

2 Research methodology & literature review

2.1 Research methodology

The goal of this paper is to minimize online money laundering by employing different data mining techniques that can predict and detect such criminal activity and assist bank experts in making the right decisions regarding clients who engage in such activity. All of which will strengthen the viability of financial institutions and boost national economies. Therefore, a search was conducted through the most popular databases, Science Direct, Springer, and IEEE, to find articles that were relevant to the paper's goal. Only articles from academic journals released between 2018 and 2024 have been considered.

The search strategy was performed based on the main terms of the paper's goal to ensure that the material employed was related to it and served as a solid basis for our paper. Thus, the search terms that were utilized were "anti-money laundering and data mining", "data mining and online money laundering," "clustering and anti-money laundering," "classification and anti-money laundering," and "association and anti-money laundering." As illustrated in Figure 1.

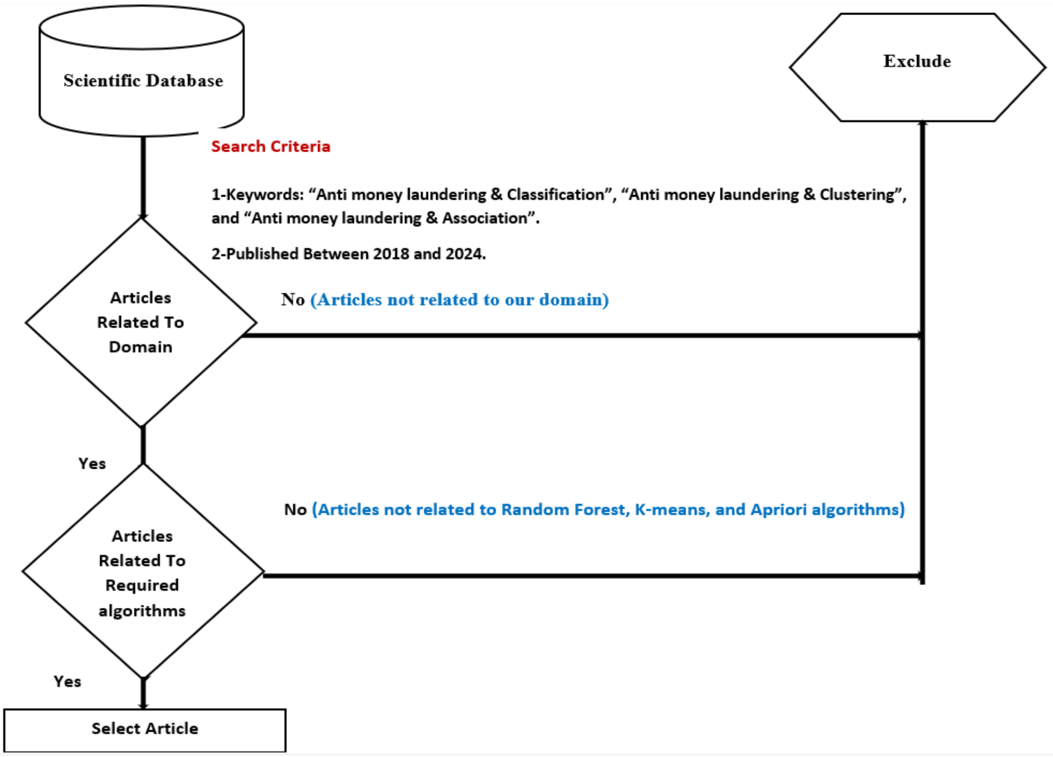

Scientific Database
Exclude
Search Criteria
1-Keywords: "Anti money laundering & Classification", "Anti money laundering & Clustering", and "Anti money laundering & Association".
2-Published Between 2018 and 2024.
Articles Related To Domain
No (Articles not related to our domain)
Yes
Articles Related To Required algorithms
No (Articles not related to Random Forest, K-means, and Apriori algorithms)
Yes
Select Article

Figure 1: Research methodology.

The search yielded 123 articles. The search results are displayed according to the year of publication in Figure 2.

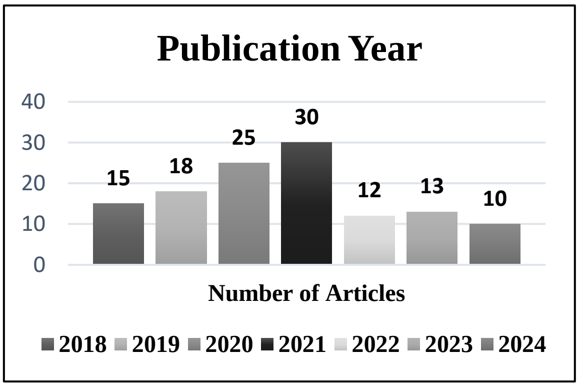


Figure 2: Classification of papers by year of publication.

2.2 Literature review

2.2.1 Online money laundering

The numerous obstacles that banks face today have an impact on their ability to compete, how they distribute their resources, and how they usually develop their strategic objectives. Online money laundering is one of these difficulties. A plethora of scholarly literature exists regarding online money laundering and the several approaches that mitigate this type of illicit activity.

[8] Provided a comprehensive review of data mining algorithms and methods applied to identify suspicious transactions. And introduced methods for anti-money laundering like link analysis, behavioral modeling, risk scoring, anomaly detection, and geographic capability. Also discussed were the essential steps of data preparation, data transformation, and data mining techniques.

[9] Discussed how data mining methods like association, clustering, and classification may be used to recognize and avoid fraud in the banking industry. The findings showed that a wide range of algorithms may be applied to fraud detection. And each of which has benefits as well as drawbacks.

[10] Demonstrated how link analysis may be applied to detecting suspicious bank transactions. And proposed a framework that includes four phases: task analysis, system design, implementation, and testing for detecting money laundering.

[4] Provided an overview of online money laundering, identifying the various anti-money laundering methods and their efficacy, the impact of online money laundering on other industries and the economy, the new avenues for online money laundering, and the detection of money laundering. The results showed that most of the research concentrated on detecting instances of online money laundering through the application of technology such as machine learning and data mining.

[11] Investigated how data mining techniques enable banks to predict and detect online money laundering. The findings indicated that due to the unavailability of high-quality datasets regarding online money laundering, there is limited scope for using supervised learning and ignorance of the key roles of unsupervised learning in detecting online money laundering.

[1] Provided a comprehensive analysis of the most recent studies on financial fraud detection, spanning the years 2009 to 2019, and categorized according to the forms of fraud and data mining techniques used. A sample of 75 pertinent papers that fell into four primary categories—bank fraud, insurance fraud, financial statement fraud, and cryptocurrency fraud—was produced because of the review. The results showed that many data mining techniques were applied in different financial applications to detect fraud.

[12] Identified data mining techniques, such as clustering, classification, and association, for financial fraud detection (FFD) and anti-money laundering (AML). And highlight how they help anticipate and identify online money laundering.

[13] Explained the data mining phases—data extraction, data cleaning and integration, data transformation, pattern discovery, pattern analysis, and information presentation—that were used to predict and detect online money laundering. Additionally, it covered numerous methods and techniques that can be applied at every phase.

[14] Proposed a system for detecting online payment fraud that makes use of data mining algorithms including random forests, logistic regression, and decision trees. The finding shows that the suggested system helps large organizations analyze the vast volume of transaction data to look for anomalies or potential fraud.

[15] Proposed a classification-based automated secure computing system for fraud detection. According to the results, the system helped process and analyze transaction data securely, look for anomalies, and notify users of potentially fraudulent activity.

2.2.2 Data mining techniques used to minimize money laundering

Analyzing and mining clients' online transactions is a crucial stage for predicting and detecting online transactions related to online money laundering that requires reliability and time. As a result, a significant number of studies were done on the best ways to produce accurate results using various data mining techniques.

[16] Illustrated and applied the supervised-based classification algorithms that predict and detect fraudulent online transactions, like naïve Bayes, logistic regression, and decision tree classifiers. The results showed that decision trees outperformed the other classifiers with an accuracy of 92%.

[17] Applied support vector machines to build a model that classifies the usual and unusual behavior that clients exhibit in online bank transactions. The findings demonstrate the effectiveness of the method in predicting banking fraud. Additionally, it demonstrates that it achieves better results when the splitting ratio is 80:20.

[18] Proposed a semi-supervised hybrid model to identify suspicious financial transactions and fraud that is based on association rule mining and clustering methods. The normal behavior patterns of the clients are extracted using the fuzzy clustering approach. Transactions that are

abnormal and do not fit into any of these clusters will be regarded as high-risk. Combining the outcomes of association rules derived from the apriori and clustering patterns allowed for the ultimate comprehension of a transaction. The results show that more frauds are detected faster when both rule-based and clustering-based components are included.

[19] Employed naïve Bayes, logistic regression, support vector machines, and k-nearest neighbors to estimate the likelihood of fraud in online banking transactions. The findings indicated that the support vector machine outperformed the other algorithms with an accuracy of 91%.

[20] Outlined the various clustering approaches, such as partitioned clustering and hierarchical clustering, that can be used to determine if a transaction is related to money laundering or not. It demonstrated how several algorithms for clustering, like sequential clustering, simultaneous clustering, divisive clustering (top-down), and agglomerative (bottom-up) clustering, can identify instances of money laundering. The results show that because simultaneous clustering is more informative, has fewer parameters, is adaptable, and can successfully integrate row and column information, it is a good fit for high-level data analysis of suspected money laundering operations.

[21] Applied frequent pattern mining to detect money laundering, the results showed that, despite the time it requires, it is very important and should not be neglected.

[22] Provided a comprehensive review of studies on the application of data mining techniques in the battle against money laundering. The results of this review demonstrated that academics were more focused on identifying suspicious transactions and that data mining techniques such as clustering and classification were applied more widely to identify money laundering.

[23] Analyzed the studies conducted on the issue of fraud detection focused on discovering indications of money laundering by applying data mining techniques like clustering and classification. The findings showed how useful algorithms for data mining are for identifying anomalous behavior and patterns of money laundering.

[24] Provided an overview of the application of classification methods like logistic regression, random forests, and support vector machines to identify online money laundering under two sampling techniques: under- and over-sampling. The effectiveness of the various approaches was evaluated using real transaction data that was supplied by a financial institution in the United States. The results showed that Random Forest outperformed the other methods.

[24] Utilized the naive Bayes classifier to predict money laundering in Bangladesh. The outcomes show that naive Bayes is a straightforward supervised learning algorithm that can classify large data sets with excellent accuracy and mobility. It is also fast, accurate, and dependable.

[26] Discussed the benefits of several supervised and unsupervised methods for anticipating and identifying online money laundering. Additionally, it demonstrated the various classification techniques and how they are used to predict suspicious transactions, including logistic regression, decision trees, random forests, and support vector machines. It also discusses how the apriori algorithm works to detect occurrences of online money laundering.

[27] Outlined the function of k-means clustering in spotting dubious clients. The findings show that it was successful in identifying clients in bank fraud situations like fishing, vishing, and skimming.

[28] Proposed an anti-money laundering system that uses the Apriori algorithm to determine the laundered money's traversal path. The findings demonstrate that the Apriori algorithm was successful in identifying odd patterns that might point to the possibility of fraud by utilizing location scanning and user behavior.

[29] Utilized historical data along with logistic regression to estimate the likelihood of money laundering. The customer’s level of risk is considered the dependent variable (high or low), and the contracted product variables (legal entity, origin, economic activity, seniority) are considered the predictors. The findings indicate that logistic regression achieved an accuracy of 89%.

[30] Applied two classification algorithms, RF and SVM, to detect and predict money laundering. The results show that the RF classifier outperforms the SVM.

[31] Utilized the gradient boosting method to classify online transactions into one of the following classes: legitimate transactions that follow the law, transactions that are marked as suspicious by a built-in alert system, and possible money laundering cases that are reported to the authorities. The results show that gradient boosting achieved an AUC of 90%.

[32] Applied decision tree and support vector machine classifiers to forecast online money laundering activity. The Python programming language has been used for the analysis. The outcome demonstrated that in terms of accuracy, recall, and precision, the decision tree performed better than the support vector machine.

[33] Applied four classification algorithms—KNN, RF, LR, and SVM—to detect and predict whether the transaction was fraudulent or not. The results show that the RF classifier outperforms the other classifiers with an AUC of 93.92%.

[7] Introduced a prediction model based on naive Bayes, logistic regression, and random forest classifiers to determine whether money laundering is likely to occur in banks. The data were derived from a simulation of actual transactions at Middle Eastern banks that were suspected of money laundering. The results indicate that the random forest classifier was found to be the best at predicting transactions related to money laundering in the bank sector.

[34] Proposed a model that uses the gradient boosting algorithm to detect and predict the existence of abnormal behavior in online transactions in the financial sector. The results indicate that the model achieved an accuracy of 93% in predicting abnormal behavior.

[35] Proposed a model that classifies online transactions as fraud or not based on DT and KNN classification methods. The results indicate that KNN outperformed DT with an accuracy of 90%.

[36] Proposed a model based on the KNN algorithm to combat online money laundering. The results indicate that the model achieved an accuracy of 93% in detecting money laundering.

[37] Demonstrated how well-supervised machine learning algorithms work for spotting money laundering activity in financial institutions. K-nearest neighbors, gradient boosting, and random forest were applied. The accuracy, precision, recall, fi-measure, and area under the curve of each classifier were evaluated to test the performance of each classifier. The findings demonstrate that the Random Forest classifier produced the best result with 93% accuracy.

Table 1 summarizes researchers' work in the domain of money laundering and illustrates the different data mining techniques used, dataset characteristics, performance metrics, and key findings.

Table 1: A summary of researchers' work in the field of money laundering.

Authors	publication year	Techniques used	Dataset characteristics	Performance metrics	Key findings
[16]	2018	Classification	A dummy dataset was used to evaluate the integrity of the data for online fraud detection.	Accuracy “92%”	A decision tree algorithm is more effective in predicting fraudulent online transactions than naive Bayes and logistic regression.
[17]	2018	Classification	The German and Australian databases of credit card datasets were used.	Precision “80%”	A support vector machine is effective in predicting usual and unusual behavior in online bank transactions under a splitting ratio of 80:20.
[18]	2018	Clustering Association	An Iranian bank dataset that includes card data from February 2015 to January 2016 was used. Each transaction has 12 raw attributes which are transaction id, time, account number, card number, transaction type, entry mode, amount, merchant code, merchant group, gender, age, and bank.	AUC “85%”	More frauds are detected faster when both rule-based and clustering-based components are included.
[19]	2019	Classification	The fraud transactions log file and the all-transactions log file were the two data sources that were combined to form the dataset. All instances of online credit card fraud are contained in the fraud transactions log file, and all transactions recorded by the relevant bank within a given time frame are contained in all transactions log files.	Accuracy “91%”	A support vector machine algorithm is more effective in predicting fraudulent online transactions than naive Bayes, logistic regression, and k-nearest neighbors.
[20]	2019	Clustering	-	-	Compare different types of clustering and their role in identifying money laundering.
[21]	2019	Association	A bank transaction dataset is used to identify money laundering transactions.	Accuracy “91%”	Graph matching algorithm is faster than frequent mining
[22]	2019	Classification Clustering	-	-	Provide literature about the role of classification and clustering in predicting and detecting money laundering.

[23]	2019	Classification Clustering	-	-	Provide literature about the role of classification and clustering in predicting anomalous behavior
[24]	2019	Classification	A dataset containing roughly a hundred explanatory variables was used, based on rich information on customer/account characteristics, recent transactions, and alert management outcomes if applicable.	AUC “91%”	A random forest algorithm is more effective in predicting fraudulent online transactions than support vector machine, and logistic regression.
[25]	2020	Classification	A bank dataset that contains attributes like "job category", "net income", "open mode", "VT", "NT", "VCT", "NCT", and "Risk class" was used.	Accuracy “78%”	A naive Bayes algorithm is fast and dependable in classifying money laundering transactions.
[27]	2020	Classification Association	-	-	Provide literature about the role of classification and association in predicting and detecting money laundering.
[28]	2020	Clustering	-	-	Identify the role of the k-means algorithm in fraud detection in banks.
[29]	2020	Association	A credit card dataset was used.	Accuracy “88%”	Identify odd patterns that might point to the possibility of fraud by utilizing location scanning and user behavior.
[30]	2020	Classification	The customer’s level of risk dataset that contains attributes of the dependent variable (high or low), legal entity, origin, economic activity, and seniority was used.	Accuracy “89%”	Identify the role of the logistic regression algorithm in fraud detection in banks.
[31]	2020	Classification	“Synthetic Financial" Datasets for Fraud Detection were used. The synthetic dataset consists of several features, including the type of transaction: CASH_IN, CASH_OUT, DEBIT, PAYMENT, and TRANSFER.	Accuracy “93%”	A random forest algorithm is more effective in predicting fraudulent online transactions than a support vector machine.
[32]	2021	Classification	An alerted transactions dataset spanning from 1 April 2014 to 31 December 2016 was used.	AUC “90%”	Identify the role of gradient boosting algorithm in predicting money laundering.
[33]	2022	Classification	A dataset downloaded from the UCI Machine Learning repository was used. This dataset contains a total of 699 instances and each instance consists of 11 attributes.	Accuracy “92%” Precision “94%” Recall “94%”	A decision tree algorithm is more effective in predicting fraudulent online transactions than a support vector machine.
[7]	2022	Classification	A bank's clients' online transaction dataset that includes 10,000 records of legitimate and fraudulent	AUC “93.92%”	A random forest algorithm is more effective in predicting fraudulent online transactions than support vector machines,

			online transactions made by clients between January 1 and December 31, 2020, was used.		K-nearest neighbors, and logistic regression.
[34]	2023	Classification	A simulated dataset of money-laundering activities in Middle Eastern banks based on a real dataset was used.	Accuracy "77%"	A random forest algorithm is more effective in predicting money laundering transactions than naïve Bayes and logistic regression.
[35]	2023	Classification	A credit card dataset was used.	Accuracy "93%"	Identify the role of gradient boosting algorithm in predicting money laundering.
[36]	2023	Classification	A banking fraud transactions dataset was used.	Accuracy "90%"	A K-nearest neighbor is more effective in predicting money laundering transactions than a decision tree.
[26]	2023	Classification	A money laundering transactions dataset was used.	Accuracy "54%"	A Part is more effective in predicting money laundering transactions than ZerO, and OneR.
[37]	2023	Classification	The Saudi General Organization for Social Insurance dataset was used.	Accuracy "90%"	A random forest algorithm is more effective in predicting money laundering transactions than gradient boosting and K-nearest neighbors.
[14]	2024	Classification	A fraud transactions dataset was used.	Accuracy "93%"	A random forest algorithm is more effective in predicting money laundering transactions than logistic regression.
[15]	2024	Classification	A credit card transactions dataset was used.	Accuracy "93%"	Proposed classification-secure fraud detection system that analyzed data securely.

According to the related research, the majority of online money laundering prediction and detection tactics use many techniques to assist banks in identifying this illicit activity, but none of the studies provide all of them, as Table 2 illustrates.

Table 2 demonstrates, most researchers used two or more data mining approaches, but no one used them all to identify, minimize, and overcome the limitations of each technique. Therefore, the main goal of this paper is to propose and implement a paradigm based on data mining techniques like classification, clustering, and association to predict and detect criminals who engage in online money laundering and mitigate the detrimental effects of online money laundering on the country's economy. In addition, as illustrated in Table 2, the three algorithms that proved their power in predicting suspicious transactions are random forest for classification, k-means for clustering, and apriori algorithm for association. Therefore, the proposed paradigm will be based on these three algorithms as illustrated in the next section.

3 A proposed paradigm (appd-oml) for predicting and detecting online money laundering

To effectively predict and detect online money laundering, the anti-money laundering methods must be updated efficiently to address online money laundering, which increases daily. Therefore, a paradigm (APPD-OML) that is based on data mining is proposed, for restructuring the process of anti-money laundering. Using APPD-OML, the information provided enables banks to easily predict and detect suspicious transactions that are associated with online money laundering.

Table 2: A summary of the data mining techniques employed by some researchers.

Author	Classification	Clustering	Association
[16]	✓	-	-
[17]	✓	-	-
[18]	-	✓	✓
[19]	✓	-	-
[20]	-	✓	-
[21]	-	-	✓
[22]	✓	✓	-
[23]	✓	✓	-
[24]	✓	-	-
[25]	✓	-	-
[27]	✓	-	✓
[28]	-	✓	-
[29]	-	-	✓
[30]	✓	-	-
[31]	✓	-	-
[32]	✓	-	-
[33]	✓	-	-
[7]	✓	-	-
[34]	✓	-	-
[35]	✓	-	-
[36]	✓	-	-
[26]	✓	-	-
[37]	✓	-	-
[14]	✓	-	-
[15]	✓	-	-
The proposed paradigm	✓	✓	✓

Therefore, the architecture of the proposed paradigm involves three layers. The first layer is the data pre-processing layer; the second layer is the mining layer; and the third layer is the report generation layer. Figure 3 presents the layers of the proposed paradigm that will assist bank experts in recognizing instances of online money laundering that take place far from bank surveillance.

3.1 Layer 1 (data pre-processing layer)

Data pre-processing is the most important layer that prepares online transactions for further analysis. Online transactions are unstructured data that may contain missing values, duplicates, and irrelevant noise that are not suitable to be used in the mining layer. As a result, the data pre-processing layer is responsible for cleaning, integrating, converting, and minimizing data to prepare it for the mining layer. Data preprocessing is made up of three important phases: the data cleaning phase, the data transformation phase, and the data reduction phase. Each of these phases has a significant role in preparing online transactions to be accurately analyzed in the mining layer.

3.1.1 Data cleaning phase

This phase oversees identifying and resolving any data anomalies, including duplicates, and missing values. The two methods that are utilized to find such anomalies are finding records that are identical in all aspects and finding empty or Nan values. Anomalies are found and fixed using a variety of techniques. Two methods are used to fix missing values: imputation and deletion. The imputation approach is used to fill a categorical variable with the dummy variable "missing" when its percentage is greater than 50%; the mode is used to fill it when its percentage is less than 50%; and the deletion approach is used to remove transactions for which specific information was missing.

Conversely, duplicates are corrected by retaining the original and deleting all others. At the end of this phase, the data becomes cleansed, free of anomalies, and prepared for the data transformation phase.

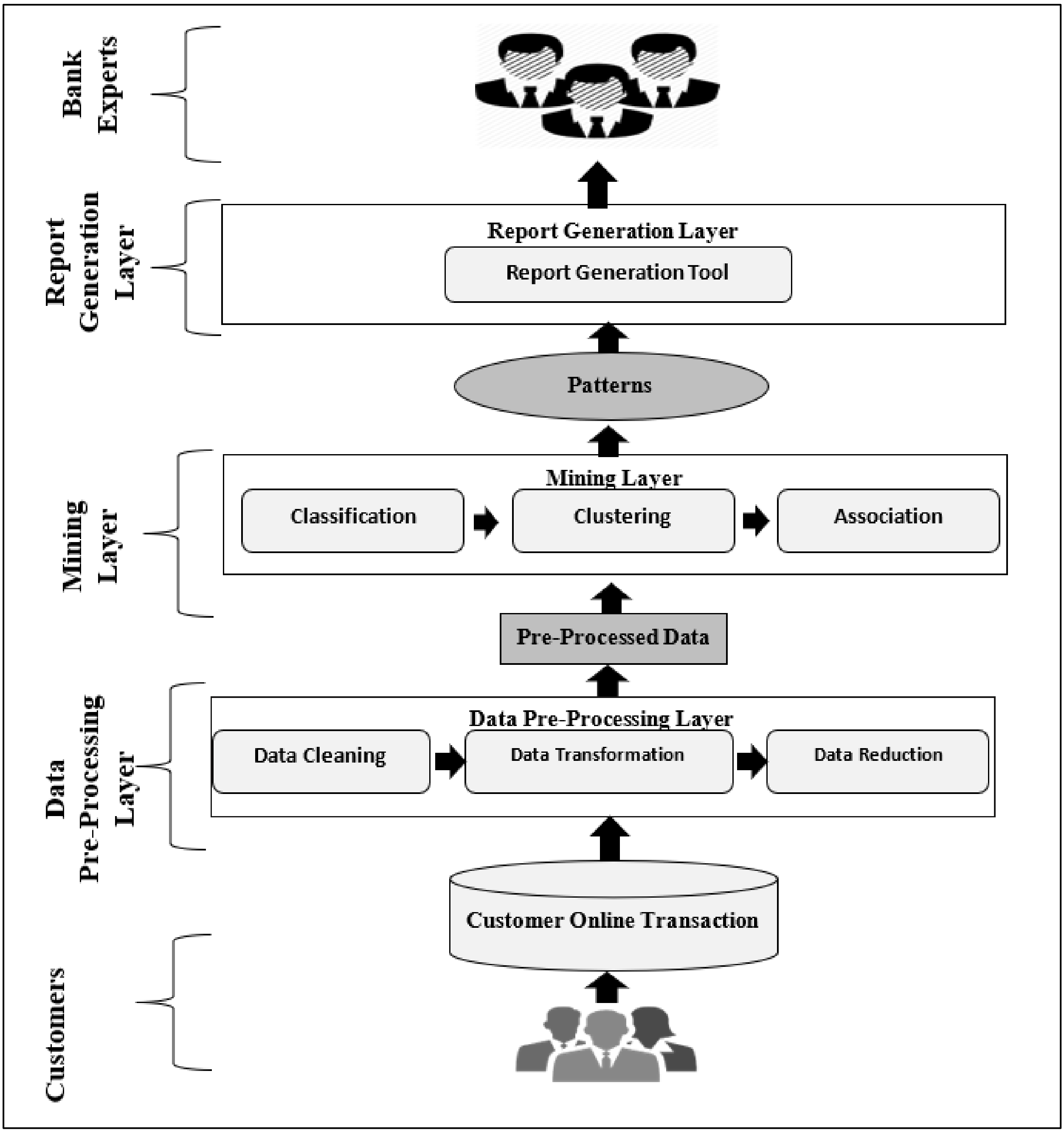


Figure 3: The proposed (APPD-OML) paradigm.

3.1.2 Data transformation phase

This phase is responsible for transforming the data into appropriate forms and generating new features that are suitable for the mining layer. It includes two stages: the first stage features construction, and the second stage is discretization.

- **The feature construction stage:** Is responsible for generating new features from the existing data that can identify clients' online behavior, the timestamps decomposition approach is employed to construct features such as, "trans_date", "trans_day", "trans_month", and "trans_hour" features derived from the "trans_date_trans_time" field. The "age" feature is derived from the "dob" field by using the subtraction method. The "distance" feature which contains the distance between the customer and the merchant derived from the "lat", "long", "merchant_lat", and "merchant_long" fields by using the calculation approach. From the "cc_num", "unix_time", and "amt" fields, the "history_30" feature is derived that contains the last 30-day spending of each credit card by applying the combination approach. From the "history_30" and "amt" fields, the "interaction_30" feature is derived by applying a combination, which contains the ratio of the current purchase price to the amount spent in the last 30 days.

- **The discretization stage:** this is responsible for dividing the range of values into intervals and placing each data point in the proper bin, the width binning approach in Equation 1 [38] is used.

$$W= \frac{\text{max}-\text{min}}{\text{Number of bins}} \qquad (1)$$

Where
W represents the width binning that has an equal width.
max is the maximum value.
min is the minimum value.
Number of bins represents the total number of bins.

At the end of the data transformation phase, all the features that demonstrate client online behavior were created and became ready for the data reduction phase.

3.1.3 Data reduction phase

This phase is responsible for eliminating unnecessary data that didn't provide any information in the mining layer while preserving important data. This is done to improve the efficiency of data analysis and avoid overfitting the model. Furthermore, this phase is also responsible for converting data to a format that can suit data mining techniques. As a result, two approaches were used in this phase: feature selection and feature extraction.

- **Feature selection:** This is responsible for preserving only important features that are used in the mining layer and eliminating other features. In online transactions, many features exist that don't add any value for predicting whether the transaction is related to online money laundering or not. In addition, some features can be identified from another feature, such as a "city" that can be identified from the client's "lat" and "long" fields. So, the existence of such features can consume a lot of processing time and doesn't add value. As a result, the feature selection approach is used to select the most significant features that can be used in the predictive model. The correlation method in Equation 2 [39] is used to measure the correlation between features and target features. Only features with a high correlation will be kept, and the others will be eliminated.

$$r = \frac{\boldsymbol{n}(\sum \boldsymbol{xy})-(\sum \boldsymbol{x})(\sum \boldsymbol{y})}{[\boldsymbol{n}\sum \boldsymbol{x}^2-(\sum \boldsymbol{x})^2][\boldsymbol{n}\sum \boldsymbol{y}^2-(\sum \boldsymbol{y})^2]} \qquad (2)$$

Where

r represents the Pearson correlation coefficient.

n is the number of values or elements.

$\sum \boldsymbol{x}$ is the sum of the 1st values list.

$\sum \boldsymbol{y}$ is the sum of the 2nd values list.

$\sum \boldsymbol{xy}$ is the sum of the product of 1st and 2nd values.

$\sum \boldsymbol{x}^2$ is the sum of squares of 1st values.
$\sum \boldsymbol{y}^2$ is the sum of squares of 2nd values.

- **Feature extraction:** This is responsible for converting features such as "categories" and "clients ages" to numerical values to be used in the mining layer. The ordinal encoding method in Equation 3 [40] is employed for this task. In ordinal encoding, each distinct category is given an integer value [41]. After completing this layer, the online transactions become ready to be used in the mining layer.

$$X(m) => N(m) \qquad (3)$$

Where
X represents the categorical value of feature m.
N represents the numerical value of feature m.

3.2 Layer 2 (mining layer)

This layer is responsible for applying the different data mining techniques to the preprocessed online transactions obtained from the data pre-processing layer. Three data mining techniques, classification, clustering, and association were employed to build the prediction model that helps bank experts in the prediction and detection of online money laundering-related transactions.

3.2.1 The role of the classification technique in prediction

This phase is responsible for building the classifier that will classify online transactions whether it is related to online money laundering or not, based on the correlation between the different features that exist in the data and the target feature. In this phase, the classifier is constructed using the random forest classifier in Equation 4 [42], and 80% of the online transactions are used to train it to produce accurate results.

$$G = 1 - \sum_{i=1}^{C} (p_i)^2$$

Where
G represents the Gini index.
C is the total number of classes.
p(i) is the probability of picking the data point with the class *i*.

3.2.2 The role of the clustering technique in prediction

This phase is responsible for grouping online transactions based on the amount since there is a strong association between the amount and the fraud status. The clusters of the amount generated will be used as one of the inputs for the association technique. In this phase, the K-means algorithm in Equation 5 [43] is used to create the clusters.

(5)

$$\mathrm{J}\ (\mathrm{V}) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\left\| x_i - v_j \right\| \right)^2$$

Where
J(V) represents an objective function.
$||x_i - v_j||$ *is* the Euclidean distance between x_i and v_j.
c_i represents the number of data points in i^{th} cluster.
c represents the number of cluster centers.

3.2.3 The role of the association technique in prediction

This phase is responsible for discovering the relationship between features in the same transaction using the association technique. As the classification technique still has a percentage of error. It indicates that certain online money laundering-related transactions may have been incorrectly categorized, which poses a serious risk to the financial industry, and to the economy. As a result, the association rule was used to analyze the client's transaction and determine whether the transaction was fraudulent based on some features that are strongly correlated with the target variable, such as" categories", "amount", and "clients' age". In this phase, the apriori algorithm in Equations 6, 7, 8, and 9 [44] is used to create the rules.

$$\text{Support} = \frac{\text{Frequency(x,y)}}{\text{N}} \qquad (6)$$

$$\text{Confidence} = \frac{\text{Frequency(x,y)}}{\text{Frequency(x)}} \qquad (7)$$

$$\text{Lift} = \frac{\text{Support}}{\text{Support(x)}*\text{Support(y)}} \qquad (8)$$

$$\text{Conviction} = \frac{1-\text{Support(y)}}{1-\text{Confidence}(x \geq y)} \qquad (9)$$

Where
N represents the total number of transactions.
x represents rule antecedent.
y represents rule consequent.

After completing this layer, the necessary patterns and data that demonstrate whether the transaction is related to online money laundering or not become ready to be used in the report generation layer.

3.3 Layer 3 (report generation layer)

This layer is responsible for taking the patterns generated in the mining layer and generating reports for banks' experts. Two types of reports are generated, including informational reports that contain information about online money laundering transactions. And analytical reports that contain qualitative and quantitative information used in decision-making. By the completion of the proposed paradigm (APPD-OML), the prediction and detection processes used to identify money laundering have been modified to incorporate the ability to identify online money laundering. This fulfills the paper's goal of enhancing the prediction and detection of online money laundering, helping banks promptly identify such criminal activity, and helping governmental authorities take the appropriate action against them.

4 Experiments and results

To validate the performance of the proposed (APPD-OML) paradigm, the clients' data collected must be prepared to help banks predict and detect online money laundering and provide experts with the appropriate information that helps them make the right decision regarding these suspicious transactions.

4.1 Data acquisition

In this paper, a bank's clients' online transaction dataset [33] that includes records of legitimate and fraudulent online transactions made by clients between January 1 and December 31, 2020, was used. These data were labeled completely and are available for open research. Table 3 illustrates dataset features.

Table 3: Features description.

Data Fields	Description
trans_date_trans_time	Contains the time and date of the transaction.
cc_num	Contains the number of credit cards.
Merchant	Contain names of merchants.
Category	Contains the domain in which transactions were performed.
Amt	Contains the amount involved in the transaction
First	Contains the first name of the client who initiated the transaction.
Last	Contains the last name of the client who initiated the transaction.
Gender	Contains the gender of the client who initiated the transaction.
Street	Contains the street of the client who initiated the transaction.
City	Contains the city of the client who initiated the transaction.
State	Contains the state of the client who initiated the transaction.
Zip	Contains the zip code of the client who initiated the transaction.
Lat	Contains the latitude of the client who initiated the transaction.
Long	Contains the longitude of the client who initiated the transaction.
City_pop	Contains the number of the city's population

Job	Contains the job of the client who initiated the transaction.
Dob	Contains the birthdate of the client who initiated the transaction.
trans_num	Contains the transaction number.
unix_time	Contains the time delay between the previous and current transaction
merch_lat	Contains the latitude of the merchant where the transaction ended.
merch_long	Contains the longitude of the merchant where the transaction ended.
is_fraud	Contains whether the client is fraudulent or not.0 means he is not fraudulent and 1 is fraudulent.

As shown in Table 4, there are no missing values in the dataset.

Table 4: Missing values and their percentage.

Column name	Percentage of missing values
trans_date_trans_time	0%
cc_num	0%
Merchant	0%
Category	0%
Amt	0%
First	0%
Last	0%
Gender	0%
Street	0%
City	0%
state	0%
Zip	0%
Lat	0%
Long	0%
City_pop	0%
Job	0%
Dob	0%
trans_num	0%
unix_time	0%
merch_lat	0%
merch_long	0%
is_fraud	0%

As previously mentioned, the (APPD-OML) has 3 layers. The data preprocessing layer is the first layer executed after data acquisition and is the most complex and time-consuming. Python, a high-level, object-oriented, and interpreter-based multipurpose programming language that gives its programmers access to a wealth of resources [45], was used to implement the proposed paradigm.

4.2 Data pre-processing layer (layer 1)

This layer is responsible for the preprocessing of the data to obtain appropriate mining in layer 2. As a result, the goal of this layer is to get the input data from Table 3 ready for the following analysis. In this layer, the following phases were applied:

4.2.1 Data cleaning

Detecting and fixing anomalies in data is one of the major difficulties in data preprocessing. Anomalies such as missing values and duplicates can lead to incorrect decisions and unreliable analysis [46]. Therefore, this phase is responsible for finding and fixing such anomalies.

- **Detecting and handling missing values**

Missing values occur when no value is recorded for a particular field in an observation [47]. The percentage of missing values in each field is shown in Table 4.

- **Detecting and handling duplicates**

Duplicates refer to records that contain identical data [48]. As shown in Table 5, there are no duplicates in the data.

Table 5. Duplicate Values in Each Record.

No. of record	Duplicate values
0	False
1	False
2	False
3	False
4	False
…..	…..
9995	False
9996	False
9997	False
9998	False
9999	False

Upon completion of this phase, it was verified that there were no duplicates or missing values in the data. It's prepared to undergo the next phase, which is data transformation.

4.2.2 Data transformation

This phase is responsible for transforming the data into appropriate forms and constructing new features that are suitable for the data mining analysis that will be applied in the mining layer.

- **Feature construction**

This stage is responsible for deriving data from the raw data to transform it into information that can be used in the prediction model, the timestamps decomposition approach, the subtraction method, the calculation approach, and the combination approach are employed to construct the following features that were used in the prediction and detection of online money laundering:

- From the "trans_date_trans_time" field, four important features, "trans_date", "trans_day", "trans_month", and "trans_hour" were derived.
- From the "dob" field, the "age" feature was derived to understand the relationship between age and whether the client is fraudulent or not.
- From the "lat", "long", "merchant_lat", and "merchant_long" fields, the "distance" feature was calculated, which contains the distance between the client and merchant.
- From the "cc_num", "unix_time", and "amt" fields, the "history_30" feature was derived that contains the last 30-day spending of each credit card.

From the "history_30" and "amt" fields, the "interaction_30" feature was derived, which contains the ratio of the current purchase price to the amount spent in the last 30 days. Table 6 illustrates the output of this stage.

Table 6. The results of the feature construction stage.

category	amt	gender	is_ fraud	trans_date	trans_day	trans_ month	trans_hour	age	distance	history _30	interaction _30
misc_net	4.97	F	1	01/01/2019	Tuesday	January	0	33	0.87283	3	0
grocery_ pos	107.23	F	0	21/02/2019	Tuesday	February	1	45	0.27231	0	10
entertain ment	220.11	M	0	01/01/2019	Tuesday	January	0	53	0.975845	1	0
gas_trans port	45	M	1	01/01/2019	Tuesday	January	0	56	0.919802	0	0
misc_pos	41.96	M	0	13/05/2019	Sunday	May	1	39	0.868505	0	0

- **Discretization**

This stage is responsible for splitting continuous data into discrete intervals to format the data for further data mining analysis. In this paper, the width binning method in Equation 1 was applied to group "cust_age” into six intervals which are '10-20', '20-30', '30-40', '40-50', '50-60', and '60 and above'. Table 7 illustrates the results.

Table 7: The results of binning the customers' ages.

category	amt	gender	is_ fraud	trans_date	trans _day	trans_ month	trans_hour	cust _age	distance	history _30	interaction_30
misc_net	4.97	F	1	01/01/2019	Tuesday	January	0	30-40	0.87283	3	0
grocery_ pos	107.23	F	0	21/02/2019	Tuesday	February	1	40-50	0.27231	0	10
entertain ment	220.11	M	0	01/01/2019	Tuesday	January	0	50-60	0.975845	1	0
gas_trans port	45	M	1	01/01/2019	Tuesday	January	0	50-60	0.919802	0	0
misc_pos	41.96	M	0	13/05/2019	Sunday	May	1	30-40	0.868505	0	0

After completing the data transformation phase, the unstructured raw data was transformed into the structured format needed for the following phases. Additionally, the raw data was utilized for constructing the key features that the prediction paradigm will employ to achieve the paper's goal.

4.2.3 Data reduction

This phase is responsible for shrinking the size of the dataset while maintaining the key information. This is done to increase the effectiveness of the data mining processes and prevent the model from overfitting.

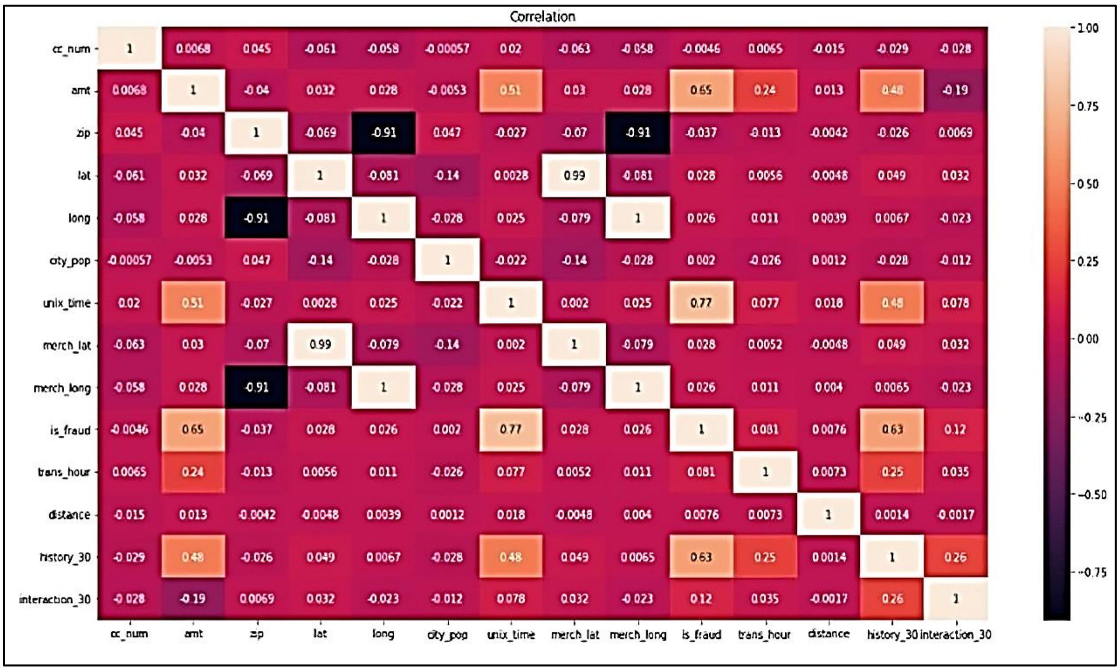


Figure 4: The correlation heatmap.

- **Feature selection**

This stage is responsible for selecting the most relevant subset of features out of the original features to be used as inputs for the mining layer. And since the utilized dataset was supervised, the filtering technique that is based on Pearson correlation in Equation 2 was used to filter the features and measure their correlation to the target feature "is_fraud”. Figure 4 presents a correlation heatmap that illustrates the correlation between the variable.

As illustrated in Figure 4, there is a multicollinearity issue that arises when two features have a correlation of more than 0.7 that must be eliminated. Furthermore, features that consumed a significant processing time and did not provide any new values to the prediction were removed. Upon the completion of this stage, as indicated in Table 8, only the features utilized for prediction remained.

Table 8: The predictive features.

category	amt	city_ pop	is_fraud	Trans _date	Trans _day	Trans _month	Trans _hour	cust_age	distance	History _30	Interaction _30
misc_net	4.97	3495	1	01/01/2019	Tuesday	January	0	30-40	0.87283	3	0
grocery_pos	107.23	149	0	21/02/2019	Tuesday	February	1	40-50	0.27231	0	10
entertainment	220.11	4154	0	01/01/2019	Tuesday	January	0	50-60	0.975845	1	0
gas_transport	45	1939	1	01/01/2019	Tuesday	January	0	50-60	0.919802	0	0
misc_pos	41.96	99	0	13/05/2019	Sunday	May	1	30-40	0.868505	0	0

- **Feature extraction**

This stage is responsible for transforming categorical data into numerical features. The ordinal encoding method in Equation 3 was used in this stage to encode features such as "category" and "cust_age".

At the end of the data preprocessing layer, the data became ready for the different data mining techniques that would be applied in the mining layer.

4.3 The mining layer (layer 2)

In this layer, different data mining techniques were applied to the preprocessed data to accomplish the paper's goal. The classification, clustering, and association techniques were used in this layer to predict and detect online money laundering.

4.3.1 Classification technique

This phase is responsible for classifying online transactions to determine if a transaction is fraudulent or not. Or to put it another way, whether it is related to online money laundering or not. In this layer, the following stages were applied:

- **Data splitting**

In this stage, an "80:20" spitting ratio was set for the classification model.

- **Build and train the model**

In this stage, the Random Forest (RF) algorithm in Equation 4 was used and was trained with 80% training data.

- **Test the model**

In this stage, the classification model was tested. The results are shown in Table 9.

Table 9: Results of the classification model.

Algorithm	Performance Measures			
	Accuracy	*Precision*	*Recall*	*F1*
RF	93%	92%	91%	91%

The results demonstrated that RF classifies online transactions with an accuracy of 93%, and an F1 measure of 91%. However, about 7% of transactions may be categorized as legitimate but include fraud. For this reason, the clustering and association techniques were applied to predict suspicious transactions based on specific features highly correlated with fraud.

4.3.2 Clustering technique

In this phase, the k-means algorithm in Equation 5 was utilized to cluster transactions according to amount since there is a high correlation between the amount and status of fraud, as illustrated in Figure 4. The elbow method was used to determine the appropriate number of clusters. As shown in Figure 5.

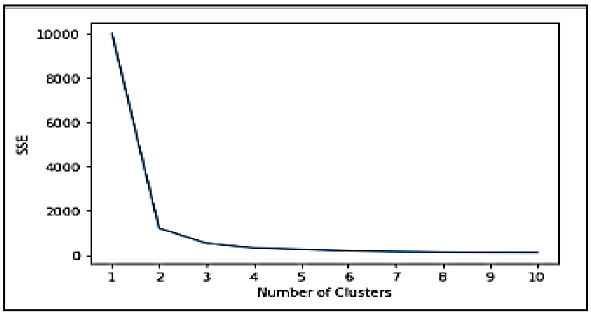


Figure 5: The results of the elbow method.

As illustrated in Figure 5, the appropriate number of clusters was two. Thus, two clusters, the "high amount" and the "low amount" were generated.

4.3.3 Association technique

In this phase, the Apriori algorithm in Equation 6 was utilized to analyze the customer's transaction and determine whether it was fraudulent based on "category", "amount", and "cust_age" features that have a high correlation to the "is_fraud" field as demonstrated in Figure 4.

A sample of the rules that were generated is illustrated in Table 10 and only rules with high confidence (>0.3) were chosen to predict whether the transaction was risky or not. The confidence of each rule represents its accuracy.

Table 10: A Sample of the generated association rules.

antecedents	consequents	antecedent support	consequent support	support	confidence	Lift	leverage	conviction
(misc_net,30-40,low_amount)	(not_fraud)	0.137856	0.245077	0.074398	0.539683	2.202098	0.040613	1.640006
(grocery_pos,40-50,low_amount)	(not_fraud)	0.245077	0.137856	0.074398	0.303571	2.202098	0.040613	1.237951
(shopping_pos,50-60,low_amount)	(not_fraud)	0.157549	0.245077	0.131291	0.833333	3.400298	0.092679	4.52954
(grocery_pos,20-30,low_amount)	(fraud)	0.245077	0.157549	0.131291	0.535714	3.400298	0.092679	1.814509
(shopping_pos,60 -Above,high_amount)	(fraud)	0.102845	0.245077	0.070022	0.680851	2.778116	0.044813	2.365427

4.4 The report generation layer (layer 3)

In this layer, two different kinds of reports were generated: analytical and informative, which provide information for banks' experts to make the right decisions regarding the different transactions that are found to be related to online money laundering.

After completing the proposed paradigm's layers, the paper's goal of predicting and detecting online money laundering that exists through online transactions such as buying and selling of goods was met, which helps banks' experts make the right decision in different situations.

5 Evaluation

To demonstrate that the (APPD-OML) paradigm outperforms the other approaches that use data mining techniques to predict and detect online money laundering various experiments were conducted that highlight the effectiveness of prediction and detection based on classification, clustering, and association. And which are made based on applying those three data mining techniques.

5.1 Experimental data

A bank's clients' online transaction dataset [33] that contains records of both legitimate and fraudulent online transactions performed by clients between January 1, 2020, and December 31, 2020, was used. These data were labeled completely and are available for open research.

5.2 The experiments methodology

The evaluation process depends on two different experiments. The first experiment tests the ability of each technique against the paradigm (APPD-OML) to achieve highly accurate results regarding the prediction and detection of online money laundering. In the second experiment, the effectiveness of the (APPD-OML) in predicting and detecting online money laundering is tested against other approaches employing the same dataset.

5.2.1 Experiment 1: (Evaluate the effectiveness of the APPD-OML paradigm against each technique)

This experiment demonstrates that APPD-OML outperforms each technique in generating highly accurate results regarding the prediction and detection of online money laundering.

To test the paradigm, in this experiment, classification, clustering, and association are applied to the data, and then the proposed paradigm is applied.

The evaluation process compares the effectiveness of the APPD-OML against each technique in providing accurate results. The APPD-OML's effectiveness and accuracy were measured in terms of accuracy in Equation 10 [49], F measure in Equation 11 [49], and AUC in Equation 12 [50]. Table 10 shows the results of Experiment 1.

- **Accuracy**

Accuracy is one of the most widely used metrics for measuring performance which is calculated as the ratio of samples that are correctly predicted to all samples

$$\text{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}} \qquad (10)$$

Where

True positive (TP): This is the number of predictions when the classifier properly identifies the positive class as positive.

True negative (TN): This is the number of predictions in which the classifier properly identified the negative class as negative.

False positives (FP): This is the proportion of predictions in which a classifier predicts a negative class as a positive one.

False negative: This is the proportion of predictions in which the classifier misinterprets the positive class as the negative class. [49]

- **F1-Score**

F1- score also known as the F-measure. It denotes the harmonic mean of recall and precision. Precision is the ratio of accurately positive samples to the total number of positive predicted samples, and recall shows the proportion of positively identified positive samples to all positive samples [49]

$$\textbf{F1-Score} = \mathbf{2} * \frac{\mathbf{Precision*Recall}}{\mathbf{Precision+Recall}} \qquad (11)$$

Where

Precision equals TP/(TP+FP).

Recall equals TP/(TP+FN).

- **AUC**

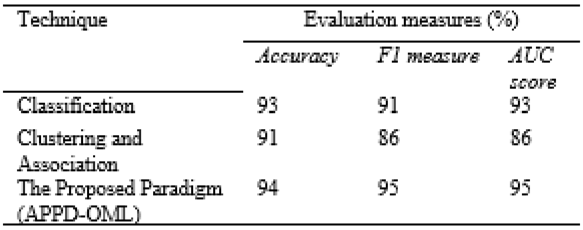
AUC is an acronym for the Area Under the Receiver Operating Characteristic Curve and is also referred to as ROC AUC. The score it produces ranges from 0.5 to 1, with 1 signifying the best outcome and 0.5 meaning the model performs as well as a chance [50].

$$\textbf{AUC} = \int \text{recall d(FPR} \qquad (12)$$

Where

FPR refers to a False positive rate and it is equal to FP/(FP+TN).

Table 11: Experiment 1 results.



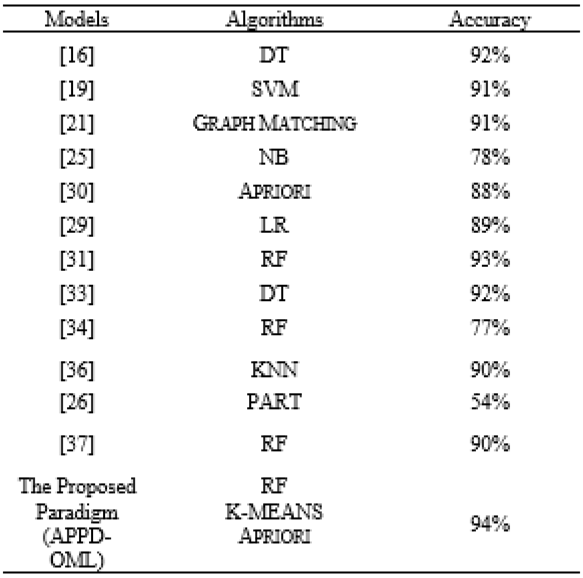
Technique	Evaluation measures (%)		
	Accuracy	*F1 measure*	*AUC score*
Classification	93	91	93
Clustering and Association	91	86	86
The Proposed Paradigm (APPD-OML)	94	95	95

According to Table 11, the findings of testing the APPD-OML indicate that the prediction based on applying the three techniques outperformed each technique with an accuracy of 94%, F1 95%, and AUC 95%. This means that the proposed paradigm outperforms each technique. In addition, it achieves a high coverage and accuracy power that is crucial for any model used in the financial industry.

5.2.2 Experiment 2: (Compare the effectiveness of the APPD-OML paradigm against the [33] approach)

The second experiment aims to compare the results of APPD-OML against [33]. The results of the proposed paradigm are compared with the ones obtained from the [33] approach and the AUC score in Equation 12 is used to evaluate the performance. Table 12 shows the results of Experiment 2.

Table 12: Experiment 2 results.



Models	Algorithms	Accuracy
[16]	DT	92%
[19]	SVM	91%
[21]	GRAPH MATCHING	91%
[25]	NB	78%
[30]	APRIORI	88%
[29]	LR	89%
[31]	RF	93%
[33]	DT	92%
[34]	RF	77%
[36]	KNN	90%
[26]	PART	54%
[37]	RF	90%
The Proposed Paradigm (APPD-OML)	RF K-MEANS APRIORI	94%

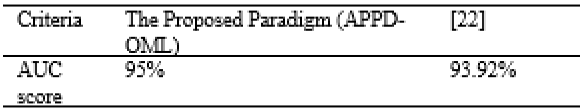
As shown in Table 12, the proposed APPD-OML outperformed the [33] approach that utilized only the classification technique in predicting fraudulent transactions.

Based on the outcomes of the two experiments, it has been proven that the APPD-OML succeeded in meeting the paper's goal of predicting and detecting online money laundering. Additionally, it decreases the effort that banks exert to monitor online transactions.

6 Discussion

In the previous sections, the impacts of various prediction and detection techniques on the online bank transaction dataset were presented. The findings show that the accuracy of prediction differs concerning the various techniques applied. As shown in Table 13.

Table 13: Comparison between different models used in prediction and detection of OML.



Criteria	The Proposed Paradigm (APPD-OML)	[22]
AUC score	95%	93.92%

In Table 13, the proposed paradigm outperformed the other models in terms of accuracy of prediction and detection, due to the application of classification, clustering, and association. In addition, it demonstrated that some algorithms are more powerful in predicting malicious activities than others.

The research results were analyzed as presented. The statistical data produced by the analysis was used to describe the findings. The results as reported are reviewed in this part with the findings of other relevant studies. The main objective of this paper is to increase the accuracy of prediction and detection of online money laundering activities. The results revealed that applying classification, clustering, and association techniques to data increases the accuracy of prediction and detection and overcomes the limitations of each technique in the banking sector**.**

7 Conclusion and future work

This paper provides a paradigm that predicts and detects potential instances of online money laundering in any nation. The proposed paradigm combines three data mining techniques to efficiently predict such criminal activity and promptly deliver the appropriate results to government authorities to take the right action against those criminals. The results of testing the APPD-OML demonstrate that it outperforms each data mining technique implemented individually in terms of accuracy and coverage; in addition, it assists banks' experts in monitoring and analyzing online transactions more efficiently. There are numerous ways to conduct further research. The following list includes a few of these directions:

- Our proposed paradigm can be extendable for minimizing online money laundering by using AI techniques such as artificial neural networks. The artificial neural network is a subset of machine learning models that are constructed using connectionism's discoveries of the principles of neuronal organization in the biological neural networks that make up animal brains, which helps in recognizing patterns in massive amounts of data across various formats, which makes it ideal for identifying suspicious transactions and risks. Previous studies ignored many artificial intelligence techniques despite their advantages. As a result, our future work was to integrate artificial neural networks into our paradigm to enable banks to effectively uncover many online transactions with various formats.
- Constructing a tool that can speed up the processing of data in the data preprocessing phase.

References

[1] Al-Hashedi, K. G. and P. J. C. S. R. Magalingam (2021). "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019." 40: 100402. https://doi.org/10.1016/j.cosrev.2021.100402

[2] Pour, M. S., et al. (2023). "A Comprehensive Survey of Recent Internet Measurement Techniques for Cyber Security." 103123. https://doi.org/10.1016/j.cose.2023.103123

[3] Wronka, C. J. J. o. M. L. C. (2022). “Cyberlaundering”: the change of money laundering in the digital age." 25(2): 330-344. https://doi.org/10.1108/jmlc-04-2021-0035

[4] Tiwari, M., et al. (2020). "A review of money laundering literature: the state of research in key areas." 32(2): 271-303. https://doi.org/10.1108/par-06-2019-0065

[5] Anwar, M. J. E. A. J. o. M. R. (2023). "The Urgency of Reforming Regulations for Money Laundering in the Digital Era." 2(7): 2895-2906. https://doi.org/10.55927/eajmr.v2i7.5009

[6] Sain, P. and S. Puri (2018). Detection of money laundering accounts using data mining techniques, Mar. https://www.researchgate.net/profile/Shalini-Puri-3/publication/323834276_Detection_of_money_laundering_accounts_using_data_mining_techniques/links/5cf45d4e92851c4dd0240b3a/Detection-of-money-laundering-accounts-using-data-mining-techniques.pdf

[7] Lokanan, M. E. J. J. o. A. S. R. (2022). "Predicting money laundering using machine learning and artificial neural networks algorithms in banks." 1- 25. Lokanan, M. E. J. J. o. M. L. C. (2019). "Data mining for statistical analysis of money laundering transactions." 22(4): 753-763. https://doi.org/10.21203/rs.3.rs-2161095/v1

[8] Chen, Z., et al. (2018). "Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review." 57: 245-285. https://doi.org/10.1007/s10115-017-1144-z

[9] Rambola, R., et al. (2018). Data mining techniques for fraud detection in banking sector. 2018 4th International Conference on Computing Communication and Automation (ICCCA), IEEE. https://doi.org/10.1109/ccaa.2018.8777535

[10] Singh, K. and P. J. I. J. o. A. I. S. Best (2019). "Anti-money laundering: Using data visualization to identify suspicious activity." 34: 100418. https://doi.org/10.1016/j.accinf.2019.06.001

[11] Canhoto, A. I. J. J. o. b. r. (2021). "Leveraging machine learning in the global fight against money laundering and terrorism financing: An affordances perspective." 131: 441-452. https://doi.org/10.1016/j.jbusres.2020.10.012

[12] Goecks, L. S., et al. (2022). "Anti-money laundering and financial fraud detection: A systematic literature review." 29(2): 71-85. https://doi.org/10.1002/isaf.1509

[13] Shrivastava, A., et al. (2023). "Literature Review on Tools & Applications of Data Mining." https://doi.org/10.26438/ijcse/v11i4.4654

[14] Tawde, S. D., et al. (2024). Online Payment Fraud Detection for Big Data. International Conference on Distributed Computing and Intelligent Technology, Springer. https://doi.org/10.1007/978-3-031-50583-6_22

[15] Singh, K., et al. (2024). "Automated Secure Computing for Fraud Detection in Financial Transactions." 177-189. https://doi.org/10.1002/9781394213948.ch9

[16] Yee, O. S., et al. (2018). "Credit card fraud detection using machine learning as data mining technique." 10(1-4): 23-27. https://jtec.utem.edu.my/jtec/article/view/3571

[17] Gyamfi, N. K. and J.-D. Abdulai (2018). Bank fraud detection using support vector machine. 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE. https://doi.org/10.1109/iemcon.2018.8614994

[18] Kargari, M. and A. Eshghi (2018). A model based on clustering and association rules for detection of fraud in banking transactions. Proceedings of the 4th World Congress on Electrical Engineering and Computer Systems and Sciences EECSS, vol. MVML. https://doi.org/10.11159/mvml18.104

[19] Thennakoon, A., et al. (2019). Real-time credit card fraud detection using machine learning. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE. https://doi.org/10.1109/confluence.2019.8776942

[20] Lokanan, M. E. J. J. o. M. L. C. (2019). "Data mining for statistical analysis of money laundering transactions." 22(4): 753-763. https://doi.org/10.1108/jmlc-03-2019-0024

[21] Dias, L. F. C. and F. S. Parreiras (2019). Comparing data mining techniques for antimoney laundering. Proceedings of the XV Brazilian Symposium on Information Systems. https://doi.org/10.1145/3330204.3330283

[22] Sobreira Leite, G., et al. (2019). "Application of technological solutions in the fight against money laundering—A systematic literature review." 9(22): 4800. https://doi.org/10.3390/app9224800

[23] Chuparkoski, D. (2019). DATA MINING TECHNIQUES FOR ANTI MONEY LAUNDERING. Proceedings of the International scientific and practical conference “Bulgaria of regions”. https://science.uard.bg/index.php/regions/article/view/602

[24] Zhang, Y. and P. J. C. E. Trubey (2019). "Machine learning and sampling scheme: An empirical study of money laundering detection." 54: 1043-1063. https://doi.org/10.1007/s10614-018-9864-z

[25] Islam, M. A., et al. (2020). "EVALUATION OF MONEY LAUNDERING RISK OF BANK ACCOUNTS USING NAIVE BAYES

CLASSIFICATION." 15(5): 3481-3493. https://jestec.taylors.edu.my/Vol%2015%20issue%205%20October%202020/15_5_43.pdf

[26] Ahluwalia, A., et al. (2023). Money Laundering Fraudulent Prediction Using Classifiers. 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE. https://doi.org/10.1109/esci56872.2023.10099770

[27] Ghosal, A., et al. (2020). "A short review on different clustering techniques and their applications." 69-83. https://doi.org/10.1007/978-981-13-7403-6_9

[28] Bagle, A. A., et al. (2020). "Anti-Money Laundering System to Detect Suspicious Account." https://d1wqtxts1xzle7.cloudfront.net/62033068/IRJET-V7I111620200208-19382-jra341-libre.pdf?1581157745=&response-content-disposition=inline%3B+filename%3DIRJET_Anti_Money_Laundering_System_to_De.pdf&Expires=1721241633&Signature=G8Ryg1itdocFq9CaeEG1THOzYiw9Vmfp6KnAFXJM3rr8nuXJWaAyJP89Qmha5q4DG6meajYSAv7uClDkZHraC9W7h2moWH3NFQevqz-xn2Yx3IKE57tmA9c0~w63R-dG5MHYgO3HlP6pat2a51o0SrmHuyytagA9Qb-BplIT0vf~kTJR8s5M5NEpEXlNmro2M3Q-7M-8ds9QRUYenxgjdoa4VcxlomPEeszWs~zCPwKlOdvtFTw6LAP45DVzrvascDPvJmK33h0Mn6aBKUnLcSNjWWnVQVcb~u6kG3e-KoFGymxXlVicXG~0t9psDALauvDUZHdL1QCfPrKZUlU1iA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[29] Martínez-Sánchez, J. F., et al. (2020). "Money laundering control in Mexico: a risk management approach through regression trees (data mining)." 23(2): 427-439. https://doi.org/10.1108/jmlc-05-2022-0061

[30] Vu, H. J. C. s. M. "Machine learning in money laundering detection." 59. https://www.researchgate.net/profile/M-Afzal-Upal/publication/341277472_Proceedings_of_the_Third_Annual_Great_Lakes_Data_Science_Symposium/links/5eb8055d4585152169c14c52/Proceedings-of-the-Third-Annual-Great-Lakes-Data-Science-Symposium.pdf#page=67

[31] Jullum, M., et al. (2020). "Detecting money laundering transactions with machine learning." 23(1): 173-186. https://doi.org/10.1108/jmlc-07-2019-0055

[32] Kumar, V. J. N. C. and Applications (2021). "Evaluation of computationally intelligent techniques for breast cancer diagnosis." 33(8): 3195-3208. https://doi.org/10.1007/s00521-020-05204-y

[33] He, Y. J. H. i. S., Engineering and Technology (2022). "Machine Learning Methods for Credit Card Fraud Detection." 23: 106-110. https://doi.org/10.54097/hset.v23i.3204

[34] Bakhtiari, S., et al. (2023). "Credit card fraud detection using ensemble data mining methods." 1-19. https://doi.org/10.1007/s11042-023-14698-2

[35] Esmail, F. S., et al. (2023). "Review of Loan Fraud Detection Process in the Banking Sector Using Data Mining Techniques." 14(2): 229-239. https://doi.org/10.32985/ijeces.14.2.12

[36] Hampo, J. A., et al. (2023). "A Web-Based kNN Money Laundering Detection System." 1(4): 277-288. https://doi.org/10.59324/ejtas.2023.1(4).27

[37] Alsuwailem, A. A. S., et al. (2023). "Performance of different machine learning algorithms in detecting financial fraud." 62(4): 1631-1667. https://doi.org/10.1007/s10614-022-10314-x

[38] Saraswat, P., et al. (2022). "Data pre-processing techniques in data mining: A Review." 10(1): 122-125. https://doi.org/10.55524/ijircst.2022.10.1.22

[39] Fan, C., et al. (2021). "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data." 9: 652801. https://doi.org/10.3389/fenrg.2021.652801

[40] Farahnak-Ghazani, F. and M. S. Baghshah (2016). Multi-label classification with featureaware implicit encoding and generalized crossentropy loss. 2016 24th Iranian conference on electrical engineering (ICEE), IEEE. https://doi.org/10.1109/iraniancee.2016.7585772

[41] Alexandropoulos, S.-A. N., et al. (2019). "Data preprocessing in predictive data mining." 34: e1. https://doi.org/10.1017/s026988891800036x

[42] Gupta, B., et al. (2017). "Analysis of various decision tree algorithms for classification in data mining." 163(8): 15-19. https://doi.org/10.5120/ijca2017913660

[43] Deng, Y., et al. (2020). "A study on e-commerce customer segmentation management based on improved K-means algorithm." 18(4): 497-510. https://doi.org/10.1007/s10257-018-0381-3

[44] Edastama, P., et al. (2021). "Implementation of data mining on glasses sales using the apriori algorithm." 1(2): 159-172. https://doi.org/10.34306/ijcitsm.v1i2.46

[45] Python, W. J. P. R. f. W. (2021). "Python." 24. Rambola, R., et al. (2018). Data mining techniques for fraud detection in banking sector. 2018 4th International Conference on Computing Communication and Automation (ICCCA), IEEE. https://doi.org/10.1109/ccaa.2018.8777535

[46] Ilyas, I. F. and X. Chu (2019). Data cleaning, Morgan & Claypool. https://doi.org/10.1145/3310205

[47] Yadav, N. and N. J. I. J. o. F. S. E. Badal (2020). "Data preprocessing based on missing value and

discretisation." 1(2-3): 193-214. https://doi.org/10.1504/ijfse.2020.10032758

[48] Nauman, F. and M. Herschel (2022). An introduction to duplicate detection, Springer Nature. https://doi.org/10.1007/978-3-031-01835-0_4

[49] Tharwat, A. J. A. C. and Informatics (2020). "Classification assessment methods." https://doi.org/10.1016/j.aci.2018.08.003

[50] Yang, S., et al. (2017). "The receiver operating characteristic (ROC) curve." 5(19): 34-36. https://doi.org/10.12746/swrccc.v5i19.391

Integrated Software Effort Estimation: a Hybrid Approach

Prerna Singal[1*], Prabha Sharma[1], A Charan Kumari[2]
[1]The NorthCap University Gurugram, India
[2]Dayalbagh Educational Institute, Dayalbagh, Agra, India
E-mail: prernasingal@yahoo.com, prabhasharma@ncuindia.edu, charankumari@yahoo.co.in
*Corresponding author

Keywords: Software effort estimation, Risk exposure, Risk management, Evolutionary algorithms, ABC, PSO, GLBPSO

Received: November 14, 2022

Risks associated with delivery of a software project and the effort spent on managing these risks are well researched topics. Very few have included this extra effort termed as risk exposure of a project, in the software effort estimate of a project. This research proposes to improve the accuracy of software effort estimates by integrating the risk exposure with the initial effort estimate of the project. A function to calculate integrated effort estimates has been defined and evolutionary algorithms ABC, PSO and GLBPSO have been used to optimize the MMRE. The approach has been tested on two datasets collected from industry, one for waterfall projects, and another for agile projects. For both the datasets, integrated effort estimates were more accurate on account of MMRE, standardized accuracy, effect size and R^2, than the initial effort estimates. Evolutionary algorithms also gave the optimum weight values at which the MMRE was optimal for both the datasets. These weight values determine the contribution of risk associated with each project cost factor in the risk exposure of the project. Integrated effort estimates have been found to be more accurate, reliable, and comprehensive than the initial effort estimates. Application of evolutionary algorithms help in reducing any bias in the integrated effort estimates.

Povzetek: Raziskava predlaga integrirano oceno dela pri razvoju programske opreme z upoštevanjem izpostavljenosti tveganjem in z uporabo evolucijskih algoritmov, kar izboljša tončnost ocen.

1 Introduction

Software Effort estimation is the basis of software project management. But it is also one of the most challenging aspects of software project management. For a long time now, project management experts have been looking for estimation techniques which provide comprehensive effort estimates with high accuracy, which is required for delivering a project within schedule and within a budget. Despite the ongoing advancements and research in the field of software effort estimation, Standish group [1] and the International society of parametric analysis [2] report that two-thirds of the projects face budget overruns and schedule delays. The challenge lies in the accurate projection of project cost factors during the initial stages of the project delivery, and managing the uncertainties encountered during the development of the final product [3]. Uncertainties in the project cost factors lead to various risks in the software development process, which need to be identified, managed, and controlled for a successful software project delivery.

The risks associated with a software project are due to factors like volatility in project requirements, availability of experienced personnel, ever-changing technology and many other project cost factors [4]. These project cost factors play a significant role in the effort estimation process as well as in project risk identification and management process [5]. Most often, risk planning and its management are treated as a separate activity from the software effort estimation task [6]. Risks are identified, analyzed, mitigated, and controlled but their impact on the effort estimate of a project is not considered. This might result in over-optimism and over-confidence in the software effort estimates of projects [7]. Thus, there is a need to integrate the risk management process in the effort estimation process for a more accurate, comprehensive, and fair effort estimate.

Managing risks at the project level would require effort towards identification, analysis, mitigation, and control of the risks associated with the project. Projects with a pessimistic effort estimate can see a reduction in the overall effort required for the project if risk management is done along with effort estimation. Projects where the effort estimation has been done in an over optimistic manner will need extra effort to manage the risks [7]. This effort which is required for the risk management process is referred as the risk exposure of the project [6]. This research proposes to include the risk exposure of the project in the effort estimate of the project. The effort estimate would now include the effort required for development of the project along with the effort required in the risk management process [8].

In the proposed approach, the cost of risk exposure is calculated according to the procedure defined by Kitchenham and Linkman [9] in 1997. They suggested that the uncertainties in the software development process cause inaccuracies in the software effort estimate irrespective of the effort estimation technique being used. The effort estimation is done in the beginning of the project when not many details are available regarding the various project cost factors impacting the effort estimate of the project. They have categorized the sources of estimate uncertainties into three types: measurement error, model error and assumption error. Assumption errors occur in the evaluations of the input parameters due to the inherent uncertainties associated with these parameters. This assumption error is the risk which creeps into the project when project cost factor values do not meet the assumed level. Kitchenham and Linkman [9], have linked the risk exposure of the project to the error in assumption of the project cost factor values. They have suggested to collect alternative cost values of the project cost factors along with the probabilities of not meeting the initial cost factor values. For each cost factor, two values are determined, the initial assumed value, and the alternative value that the project cost factor might attain during the execution of the project. Kitchenham and Linkman compute the effort/ cost of managing risk as follows: For each factor difference in the initial and alternate cost is multiplied by the corresponding probability of not meeting the initial cost of that factor. This is the cost of managing risk related to that factor. Sum of all these costs define the total effort/ cost for managing the risk of the whole project or the risk exposure of the project.

The authors in an earlier paper [8] have added the cost of risk exposure to the initial effort estimate of the project to obtain an integrated effort estimate$(IE_{initial})$. Equation (5) gives the formula to calculate$(IE_{initial})$.

The probabilities and project cost factor values in the formula, proposed by Kitchenham and Linkman, are based on expert judgment, which may result in a biased effort estimate. In the proposed approach, these probabilities are treated as variables and these variables/ weights are then optimized using evolutionary algorithms like Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), and a hybrid global local binary particle swarm optimization algorithm (GLBPSO), to obtain more accurate and fair effort estimates. This reduces the dependency of the model on expert judgment and removes any bias in the integrated effort estimates due to the probabilities. This research determines the optimal value of the estimated effort and the weights which determine the impact of risks due to each cost factor on the estimated effort. The proposed approach has been tested for waterfall model delivery projects as well as projects using agile delivery.

The main contributions of this research are as follows:

- Risk exposure of a project is added to the effort estimate of the project which gives an integrated effort estimate for the project.
- A weighted function for calculating the integrated effort estimate has been defined.
- Two questionnaires have been prepared: one based on CoCoMo II project cost factors for waterfall delivery projects, and another based on project cost factors by Ziauddin for agile delivery projects. Based on the responses received for these questionnaires, two datasets have been collected: ‘Waterfall model dataset’ and ‘Agile model dataset’.
- Integrated effort estimate has been optimized using evolutionary algorithms ABC, PSO and GLBPSO.

2 Related work

The need for risk assessment and risk control in software development projects was highlighted by Boehm in 1989 [10]. His work proposed a framework for identification of software validation and verification activities and the effort which would go into the risk management process. They attributed 40%-50% of the software project costs to rework costs, and 80 % of that rework costs to the highest risk factor associated with the project. Fairley in 1994, used regression cost modelling of the effort estimates of the past projects in the organization to determine the risks associated with a software project [11]. Residual error due to the difference between the estimated effort and actual effort was used to determine the project cost factors which impacted the software effort estimates. He suggested controlling and mitigating these project cost factors to manage the project risk.

In 1997, Madachy proposed a heuristic to calculate the cost of risk exposure of the project using the project cost factors [12]. He identified risk rules based on the CoCoMo cost factors to assign levels to the risks identified in the project. The cost of risk exposure of the project was calculated based on the risk level and the contributing project cost factor values. Briand et al. introduced a hybrid model for project effort estimation, risk assessment and benchmarking without relying on the historical data for project effort estimate [13]. The proposed model based its effort estimate on productivity of the project and calculated the project cost overheads based on a questionnaire, which would be filled by the experts associated with the project.

In 2006, Jantzen estimated the impact of project risks on project effort estimate, project duration and project quality. His work emphasized on re-estimating and re-planning the software project during its execution, based on the various risks, their level and risk status at various stages of the project [14]. Huang et al. in 2006, proposed an effort estimation technique based on the fuzzy and uncertain nature of the project cost factors. The effort was estimated using fuzzy decision tree where along with the effort estimate of the project, the estimation error was used for risk analysis and management [15]. Manalif in 2013, proposed a fuzzy expert – CoCoMo model which added a contingency to the estimated effort based on the project cost factors of the CoCoMo model [16]. The contingency effort was kept separate from the total effort estimate of the project.

In 2017, Aslam et al. considered the risks associated with rich mobile application development projects developed using agile methodology [17]. Along with risk factors, they also included the quality aspect of the project in the effort estimate of the project which enabled the development effort estimation at multiple quality levels. Their work was limited to project on rich mobile application development.

In 2018, Koutbi & Idri proposed inclusion of the cost of risk management in the effort estimation process at the organization level instead of handling it at a specific project level [18]. They argued that risk is better handled and mitigated over a portfolio of projects which improves the effort estimation process of the organization. On the other hand, organizations can have projects of varied nature, for which the project cost factors contributing to risks may vary.

In 2019, Ramakrishnan et al. built a multilayer perceptron model to estimate the software development effort. The model included project risk score in the effort estimation process [19]. They used an enhanced gradient boosting technique which decreased the standard deviation of the residuals indicating better effort estimation results.

In 2020, Michael Kataev et al. [20], reiterated the importance of including the cost of risk management in the software development process for on time and in budget software project deliveries. They included the internal and external risks which impacted the overall financial health of the organization.

It is clear that researchers have focused on integrating risk management process with the software effort estimation process, but none have included the effort spent on risk management in the effort estimate of the project itself. Thus, this research tries to bridge this gap and analyses the impact of risk exposure on the effort estimate of the project. Also, the probability of risk occurrence is optimized using evolutionary algorithms ABC, PSO, GLB-PSO.

Rest of the paper is organized as follows: Section 3 details the proposed approach of integrating the impact of risk on the software effort estimate and optimizing it using evolutionary algorithms. It outlines the algorithms used in the proposed approach. Section 4 gives the details of the datasets collected for this research. Section 5 contains the details of calculating integrated effort estimates for waterfall and agile projects. Section 6 gives the details of experimental setup and algorithms for ABC, PSO and GLBPSO. It contains the details of the fitness function used in the algorithms and explains the evaluation criteria used for comparing the proposed approach with the baseline effort estimation techniques. Section 7 presents and compares the experimental results obtained for both the waterfall and agile delivery projects. Section 8 points out some threats to validity of the proposed approach. Section 9 draws the conclusion, and Section 10 describes the scope for future work.

3 Theoretical background

This section explains the risk exposure of a project, research questions which motivated the research and presents the proposed model for calculating integrated effort estimates.

3.1 Risk exposure

The risk exposure of the project is the total effort required to identify, mitigate, and control the risks in the project which occur due to various project cost factors. Risk exposure due to each cost factor of the project is calculated separately and then added together to give the total risk exposure of the project. These risks arise due to the uncertainties associated with the software projects. The initial effort estimate of the project is determined based on certain assumptions made regarding the project cost factors like reliable requirements, team communication, availability of hardware and software resources, and expertise & experience level of the team. Since very little information is available when these assumptions are made at the beginning of the project, quite often these assumptions are not met giving rise to risks which may increase or decrease the effort required to develop the project.

According to Kitchenham and Linkman [9], for a project with n cost factors, initial effort estimate $(E_{initial})$ required to develop a project is calculated using the assumed project cost factor values at the beginning of the project. To calculate the risk exposure of the project, risk exposure due to each project cost factor is added. Risk exposure of the i^{th} project cost factor is calculated by multiplying the probability of not meeting the initial level $(p_{i,alter})$ of the i^{th}project cost factor with difference between the effort estimated $(E_{i,alter})$ at the alternative level of the i^{th}project cost factor and the initial estimated effort$(E_{initial})$. The risk exposure (E_{risk}) of a project can be calculated using the equation given below:

$$E_{risk} = \sum_{i=1}^{n}(E_{i,alter} - E_{initial}) \times p_{i,alter} \qquad (1)$$

This risk exposure is the extra effort that would be needed for risk management and planning of the project.

3.2 Proposed approach

In the proposed approach, the risk exposure of a project has been added to the initial effort estimated required to develop the project. Thus, according to the proposal in this research, the total effort estimate of a project must reflect the effort involved in managing the various risks encountered during the completion of the project. If $E_{initial}$ denotes the initial effort estimate of a project, E_{risk} is the risk exposure calculated using equation (1), then the integrated effort estimate $(IE_{initial})$ of the project can be determined using the equation given below:

$$IE_{initial} = E_{initial} + E_{risk} \qquad (2)$$

Substituting the value of E_{risk} in equation (2) from equation (1) will give us the initial integrated software development effort estimate for a project.

$$IE_{inital} = E_{initial} + \sum_{i=1}^{n}(E_{i,alter} - E_{initial}) \times p_{i,alter} \qquad (3)$$

This estimate will now include the effort that would be required for development as well as risk management, and planning of the project. Values of the probabilities $p_{i,alter}$ of not attaining the assumed initial level of project cost factors in equation (3) can be ascertained from the program manager or the expert responsible for effort estimation of the project. It is possible that the values of the probabilities $\{p_i\}_{i=1}^{n}$may be biased, and hence may not accurately estimate the cost of risk management.

In this paper, p_i's are treated as variables between 0 and 1 and then their optimal values are computed using evolutionary algorithms such as ABC, PSO and GLBPSO, so that the estimated cost is as close as possible to the actual cost of a project. Therefore, in equation (1), the p_i's have been replaced by w_i's. The risk exposure of a project is expressed as the weighted mean of the risk exposures due to all the project cost factors using the formula given below:

$$E_{risk} = \sum_{i=1}^{n}(E_{i,alter} - E_{initial}) \times w_i \qquad (4)$$

The formula in equation (3) for calculating integrated software development effort estimate of a project (IE) can now be expressed using the equation given below:

$$IE = E_{initial} + \sum_{i=1}^{n}(E_{i,alter} - E_{initial}) \times w_i \qquad (5)$$

The w_i in equation (5) can now be optimized using evolutionary algorithms.

In this research three evolutionary algorithms have been experimented with: ABC, PSO and GLBPSO. Previous research on software effort estimation has demonstrated promising results with the application of evolutionary algorithms such as ABC, PSO, and GLBPSO [21-23]. These algorithms have been effective in optimizing complex, non-linear software project effort estimation problem.

3.3 Artificial bee colony optimization

Artificial bee colony (ABC) is a metaheuristic algorithm proposed by Karaboga et al. [24], which is based on the intelligent social behavior of the honeybee swarm. ABC algorithm employs collaborative trial and error approach to identify honeybee swarm. The ABC optimization algorithm is driven by peer-to-peer learning behavior of social colonies and reaches the optimal solution following an iterative process. There are four phases in the ABC algorithm: initialization phase, employed bee phase, scout phase and onlooker bee phase. In the initialization of the population, ABC generates a uniformly distributed population of solutions where each solution is a dimensional vector. In the proposed approach, the weight vector $(w_i's)$ described in Section 6.1 will represent the solution vector. Each w_i varies between 0 and 1. The employed bees update the current solution based on their own experience and fitness value of the new solution. If the new solution has a higher fitness value than the current solution, the bee selects the new solution and discards the current one. In this research, weight vector which gives the lowest MMRE will be selected.

3.4 Particle swarm optimization

Particle swarm optimization (PSO) algorithm also belongs to the family of swarm intelligence algorithms, and was first proposed by J. Kennedy and R. Eberhart in 1995 [25]. PSO algorithm models the social behavior of flocking of birds or school of fish to optimize nonlinear functions. Each particle/bird which represents the solution to the problem has a position and velocity associated with it. In the algorithm, particles change their position by adjusting their velocity either to seek food, avoid predators or to identity optimized environmental parameters. Also, each particle memorizes its best position during the process and communicates it to other particles in the swarm. So, the velocity of a particle is modified using the flying experience of the particle itself and the flying experience of the whole group, termed as global best PSO. For this research, the particles are the weight vectors which determine the risk exposure due to each cost factor. The objective is to minimize the MMRE of the project in the dataset.

3.5 Global local binary PSO optimization

An improved PSO algorithm was proposed by Rita Chhikara et al. [26] to overcome the disadvantages of the global best PSO. The algorithm Global local binary PSO (GLBPSO) integrated the global best PSO with local best PSO and dynamically changed the population size using (Hope/ Rehope). The algorithm begins by using the global best PSO and if the value of the fitness function does not change for two consecutive iterations, local best PSO is applied with a neighborhood size of 4. The particles move towards the best solution in the neighborhood by communicating with their four immediate neighbors. If the local best PSO does not improve the fitness function in three consecutive iterations, this indicates stagnation in the search process. To avoid this stagnation, hope/ rehope is applied on the population. For kth iteration, for particles having marginal distance (less than 0.01) among themselves, only the particle with higher fitness function value is retained in the population. However, it could lead to decrease in population size. To avoid such a situation, the population size is checked after each iteration, and if it reduces to less than 50% of its original size, then the population size is increased randomly by 30%. This eliminates the bad performing particles and at the same time revives the hope for a better solution. These steps are repeated until the stopping criteria is met or the set number of iterations have been executed.

3.6 Research questions

This paper aims to provide the experimental evidence to answer the research questions given below:

RQ1: *Does the accuracy of effort estimate of the project improve by adding the cost of risk exposure to the initial estimated effort of the project?*

This research integrates the effort that goes into risk management and control into the initial effort estimate of the project. A function $IE_{initial}$ as given in equation (3) is proposed, to determine the integrated effort estimates for software projects. The weighted cost of the risk exposure due to each project cost factor is added to the initial effort estimate of the project. The research aims to find out whether these integrated effort estimates are more accurate than the initial effort estimates.

RQ2: *What is the impact of bias on risk exposure of the project?*

Some biases in the cost of risk exposure might have been introduced due to $p_{i,alter}$ in the integrated effort estimates $(IE_{initial})$, as given in equation (3). The values of $p_{i,alter}$ are based on expert judgment and have been collected based on a questionnaire for the respective effort estimation models: waterfall and agile. These values might be biased as per the expert's understanding and knowledge. This research reduces these biases by obtaining optimal values of the $w_i's$ $(p_{i,alter})$ by using evolutionary algorithms ABC, PSO and GLBPSO. In the weighted function $IE_{initial}$ in equation (3), the probabilities are replaced with weights to obtain the weighted function IE as given in equation (5). IE is optimized using evolutionary algorithms, which obtain optimal values of the weights $w_i's$.

RQ3: *Can project cost factors be ranked with respect to their risk exposure?*

Using equation (5), optimal values of $w_i's$ are obtained. These weights determine the contribution of each project cost factor in the total cost of risk exposure of the project. Higher the value w_i implies that the associated project cost factor will contribution more to the total risk exposure of the project. This information can be used by project managers to identify the project cost factors which need better management of the risks associated with them.

4 Data collection

The proposed integrated effort estimation model was tested on data collected from an Indian IT firm involved in software development, maintenance, and consultancy. Two types of projects were considered for the research – projects with Waterfall delivery model [27] and projects with Agile delivery model [28]. Experts who handled the projects were interviewed over a span of 1 year and data was collected based on a questionnaire. Two separate questionnaires were prepared – one for each delivery model, Waterfall and Agile. Experts included project managers, technical architects, analysts, and developers. These experts were directly involved in the project effort estimation process. Experts from over 75 different projects were interviewed. 45 projects followed the Waterfall delivery model, and rest of the 30 projects were working on the agile principles. Projects were from varied domains covering banking, healthcare & pharmaceutical, and Insurance. Waterfall model questionnaire had 69 fields to be filled while the agile questionnaire had 45 fields. Table 1 shows a general format of the questionnaire.

Table 1: Questionnaire format

Questionnaire					
Project Id	KLOC / Story Points	Actual Effort (Man Months)	Cost Factor		
			Initial Level	Probability of not meeting the initial Level	Alternative Level

Questionnaire for the Waterfall model was based on CoCoMo II project cost factors. There were 5 scale factors and 17 cost factors identified in the CoCoMo II Model. All the scale and cost factors have been calibrated at five levels: very low, low, nominal, high, very high and extra high [29]. The questionnaire focused on lines of code in the project (measured in KLOC), actual effort spent (Man Months), scale factors and the cost factors. For the scale and cost factors three inputs were taken from the experts – their initial assumed level while estimating effort, probability of not meeting that assumed level and an expected alternative level. The dataset thus collected is referred to as the "Waterfall model" dataset.

Similarly, questionnaire for the Agile model was based on the frictional and dynamic forces suggested by Ziauddin [30] . In the Agile delivery model, the stories are delivered in sprints. This questionnaire collected responses for one sprint in each project covering story size, story complexity, actual velocity, sprint time, dynamic factors, and frictional factors related to the project. Size of the story was rated on a scale of 1-5 based on the effort required for the development of the story. Complexity was also rated on a scale of 1-5 depending upon the nature of the work and complexity of technical and non-technical requirements. There are 4 frictional factors and 9 dynamic factors identified in the model which impact the effort estimates of agile projects. Ziauddin has laid down guidelines to determine the size and complexity of the story on a scale of 1-5. The questionnaire focused on the sprint time, story size, actual velocity, story complexity and the variable forces (dynamic & frictional factors) – their initial assumed levels during effort estimation, probability of not meeting that assumed level and the expected alternative level. The dataset thus collected is referred to as the “Agile model” dataset.

5 Calculation of Integrated effort estimates

During the initial project planning phase, an estimate of the effort ($E_{initial}$) involved in development of a project is done with certain assumptions regarding the project cost factor values. With these assumptions the project cost factors are assigned certain values and then the initial effort is estimated. This effort can be estimated using any established effort estimation technique depending upon the nature of the project, its delivery model and local environment. The effort calculated is expressed in Man Months (MM), which is the average effort spent by one person for a month. Initial effort is calculated for both the datasets: waterfall and agile using the estimation techniques described below.

5.1 Waterfall model dataset

For the “Waterfall model” dataset, the project data was collected based on CoCoMo II project cost and scale factors [29]. So, the initial effort values ($E_{initial}$) for a project were calculated using the CoCoMo II effort estimation formula given below:

$$E_{initial} = A \times Size^{S} \times \prod_{i=1}^{n} EM_{i,initial} \tag{1}$$

where
$$S = B + 0.01 \times \sum_{k=1}^{5} SF_{k,initial} \tag{2}$$

A is a constant whose value can be calibrated according to the project’s local environment. It has been established that CoCoMo II estimates the software development effort more accurately when the constant A is calibrated according to the organization’s productivity and activity distributions [31]. This research uses the standard value of 2.94 proposed in the CoCoMo II Model. B is also a constant set at 0.90 in the CoCoMo II model. $EM_{i,initial}$ denotes the project effort multiplier for the i^{th} project cost actor which impacts the estimated effort of the project. There are 17 cost factors (n=17) in the CoCoMo II Model. Size of the project is determined in KLOC. $SF_{k,initial}$ are the five scale factors. From the expression for S, it can be observed that that SF_k’s make the effort grow exponentially. Substituting the values of $E_{initial}$ from equation (6) in equation (5), the integrated effort estimate (IE), for Waterfall model dataset can now be expressed as below:

$$IE = A \times Size^{S} \times \prod_{i=1}^{n} EM_{i,initial} + \sum_{i=1}^{n} \left\{ \left(A \times Size^{S} \times \prod_{i=1}^{n} EM_{i,alter} \right) - \left(A \times Size^{S} \times \prod_{i=1}^{n} EM_{i,initial} \right) \right\} \times w_i \tag{3}$$

where $EM_{i,initial}$ represent the project cost factor values at the initial assumed stage, and $EM_{i,alter}$ represents the project cost factor values at the alternative level.

5.2 Agile model dataset

For the “Agile model” dataset, initial effort value $E_{initial}$ for a project was calculated using the model proposed by Ziauddin [30]. The model estimates the effort for a sprint using the story size, complexity, dynamic and frictional factors. The software product to be developed, is described in the form of user stories creating a product backlog owned by the product owner, usually a representative of the customer for whom the product is being developed. The team delivers the selected user stories at completion of each sprint. As opposed to waterfall model where the manager is responsible for estimating the effort in the planning phase, in agile approach the team members decide on the effort that will go into the delivery of the user story at the beginning of each sprint. Team members estimate the required effort based on their experience, story size, complexity, and project cost factors. The effort is expressed in terms of story points, where one story point corresponds to a day’s work for the team member. The project cost factors might change during the sprint execution leading to the uncertainty in effort estimate by the team member. These project cost factors account for the risks associated with the project which impact the effort estimate of the sprint. Steps given below were followed to calculate the initial effort estimate.

a) **Effort for a story:** For each story, the effort dispensed towards the development of the story was calculated using the formula given below:

$$ES\ (Effort\ for\ a\ story) = story\ size \times story\ complexity \tag{4}$$

This effort estimate of the story is expressed in story points.

b) **Effort for the whole sprint:** The estimated effort for all the stories in the sprint is added to get the effort estimate of the sprint, using the equation given below:

$$E\ (Effort\ for\ whole\ sprint) = \sum_{i=1}^{n} ES_i \tag{5}$$

where, n is the number of stories being delivered in the sprint. Now, the effort for the whole sprint is in story points.

c) **Variable factors**: From the agile project dataset, initial values of the frictional and dynamic factors were used to calculate the impact of variable factors on the initial effort estimate. The impact was calculated using the formula given below:

$$D\ (Variable\ Forces) = \prod_{k=1}^{4} Frictiona\ factors_i \times \prod_{m=1}^{9} Dynamic\ factors_j \tag{6}$$

d) **Agile Velocity:** In this step, the velocity for each sprint in the project was determined based on the estimated sprint effort E, sprint time T and variable forces D in the sprint, using the formula given below:

$$V\ (Velocity) = \left(\frac{E}{T}\right)^{D} \tag{7}$$

In Agile delivery, the velocity of the project is improved and stabilized over various sprints [32]. This stability in velocity will depend on the project cost factors, in this case dynamic and friction factors. These factors change often during the execution of the sprint thus leading to uncertainties in the estimated effort. These uncertainties are the risks associated with the project which need to be addressed during the project execution. The effort that goes into the control and mitigation of these risks has been accounted for in the estimated effort in the proposed approach. The formula for integrated effort ($IE_{initial}$) is given in equation (13).

e) **Effort estimate**: From equation (12), formula for $E_{initial}$ can be obtained as follows:

$$E_{initial} = E = (V)^{\frac{1}{D}} \times T \tag{8}$$

Substituting the values of $E_{initial}$ from equation (13) in equation (5), the integrated effort estimate for the Agile model dataset can now be expressed as below:

$$IE = (V)^{\frac{1}{D}} \times T + \sum_{i=1}^{n} \left\{ \left((V)^{\frac{1}{D_{i,alter}}} \times T \right) - \left((V)^{\frac{1}{D}} \times T \right) \right\} \times w_i \tag{9}$$

where $D_{i,alter}$ represents the variable forces or project cost factors at the alternative level.

6 Experimental setup

The estimated effort expressed in equations (8) and (14) for waterfall and agile delivery models respectively can now be optimised using evolutionary algorithms. In this work, ABC, PSO and GLBPSO have been used to optimise the effort estimates as described below:

6.1 Representation of the population

In this research, each individual particle/ bee in the population is represented as a vector of weight values as shown below:

Waterfall model

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{17}	w_{18}	w_{19}	w_{20}	w_{21}

Agile model

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}

The Waterfall model and the agile model datasets will have 21 and 13 weights respectively for associated project cost factors. In equations (8) and (14), the weights are optimized to calculate the integrated effort estimates for software project development. All the weights are in the range 0 to 1. The algorithms will return the vector with weight values at which the MMRE value is minimized. From equation (15), it follows that the optimization of weight values will bring the IE`s close to the actual effort.

6.2 Fitness function

In evolutionary algorithms, fitness function is used to evaluate the fitness of the individuals in the population. Mean magnitude of relative error (MMRE) is the most widely used fitness function for software effort estimation problem [33]. Tomas Urbanek [33] et al. have found MMRE to be an average fitness function for the effort estimation problem. Considering earlier performance of MMRE, we have used MMRE as the fitness function for ABC, PSO and GLBPSO algorithms.

6.2.1 Mean magnitude of relative error

For the algorithms used in this research, mean magnitude of relative error (MMRE) is used as the objective function to evaluate the fitness of the individuals in the population [34]. Magnitude of relative error (MRE) is the ratio of the absolute difference between the integrated effort (IE) and the actual effort spent on the project, and the actual effort spent on the project. Thus, the formula for MMRE will be:

$$MMRE = \frac{\sum_{i=1}^{N} MRE}{N} = \sum_{i=1}^{N} \left(\frac{IE_i - actual\ effort_i}{actual\ effort_i}\right) / N \tag{10}$$

where N is the total number of projects in the dataset. All the three algorithms are used to obtain weight values (w_i) that minimize the MMRE.

6.3 Parameter values

The proposed approach is implemented on two datasets collected during the research described in section 4. The implementation for the considered evolutionary algorithms was executed on MATLAB. The fitness function values in ABC, PSO and GLBPSO showed little change after 20 iterations. So, the number of iterations is set to 20 in all the swarm optimization algorithms.

6.3.1 Parameter values for ABC

In ABC algorithm the population size is generally kept as the square of the number of employed bees [24]. The number of weights in the waterfall model dataset is small (21), an initial population of size 441 i.e., square of 21 was generated randomly. Similarly, the agile model dataset had 13 weights, so the randomly generated population size was kept at 169. The best fitness value after every iteration showed little change after around 17 iterations as shown in figures 2 and 6. Thus, the number of iterations in each run were set to 20. The number of trials for abandoning food source (Limit) was set to 50.

6.3.2 Parameter values for PSO and GLBPSO

The population size for PSO and GLBPSO were kept the same as for the ABC algorithm. So, the population size was set at 441 for Waterfall model dataset and 169 for agile model dataset. Other parameters values are given below:

1) Number of iterations: 20
2) $c_1 = c_2 = 1.5$
3) w = 0.8

For PSO and GLBPSO, the best fitness value after every iteration showed little change in waterfall model dataset after around 17 iterations and showed little change in agile model dataset after around 7 iterations as shown in figures 3, 4, 7 and 8 respectively.

6.4 Performance evaluation metrics

The integrated effort estimated with the proposed model is compared with the initial estimated effort using the benchmark model based on four performance evaluation metrics: mean magnitude of relative error (MMRE), standardized accuracy (SA), effect size (Δ) and coefficient of determination (R^2).

6.4.1 Standardized accuracy

The performance evaluation measures MRE and MMRE have been criticized for being biased towards effort estimation techniques resulting in underestimates [34-39]. Therefore, we compare integrated effort estimates from the proposed approach with the estimated effort of CoCoMo II and Ziauddin models using standardized accuracy (SA) also. In SA, an estimated effort value is randomly chosen and assigned as effort estimate of the remaining projects. This process is repeated 1000 times and then mean absolute error (MAR) is calculated every time. Standardized accuracy is calculated based on the formula given below:

$$SA = 1 - \frac{MAR}{MAR_{P0}} \times 100 \tag{11}$$

where MAR is the mean absolute error i.e., the mean of the absolute difference between the estimated and actual effort estimates of all the projects.

MAR_{P0} is the mean MAR of the 1000 random assignments [38]. For performance evaluation, a lower MMRE value or a higher SA value implies a better effort estimation approach.

6.4.2 Effect size

Effect size (Δ) is used to determine the reliability of the proposed approach [38, 40]. It can be calculated based on the formula given below:

$$\Delta = \frac{MAR - MAR_{P0}}{\sigma_{P0}} \tag{12}$$

where σ_{P0} refers to the standard deviation of MAR values of 1000 random assignments from MAR_{P0}. High value of Δ (>0.5) indicates that the results obtained by the proposed algorithm are more reliable than those obtained by random guessing.

6.4.3 Coefficient of determination

Coefficient of determination (R^2) is used to determine the correlation between the dependent and the independent variables [41]. It varies from 0 to 1. A value closer to 1 indicates a strong correlation between the variables. For this research, independent variables are the project cost factor values and the size of the project. Estimated effort will be the dependent variable.

7 Results & analysis

Each algorithm was run 25 times. We have reported the best results obtained for each algorithm.

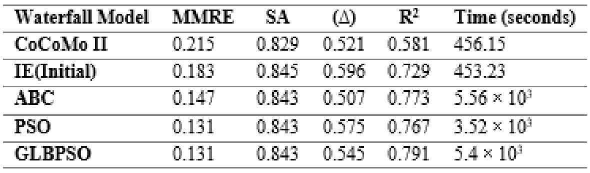
7.1 Results for waterfall model dataset

Table 2 lists the weight values obtained for all the three algorithms: ABC, PSO and GLBPSO for the Waterfall dataset. These weights represent the optimal values of $p_{i,alter}$ defined in equation (5). A higher value of $p_{i,alter}$ (>0.5) indicates high level of risk exposure due to the corresponding project cost factor. Weight values 0, indicate that there was negligible risk due to that project cost factor and it did not impact the effort estimate of the project. All the projects considered are from the same organization, where these project cost factors such as main storage constraint, platform volatility, platform experience and execution time constraint, might already be controlled efficiently thus having no impact on the integrated effort estimate. Weight values of 1 indicate that the associated project cost factors highly impact the integrated effort of the project. These factors need to be monitored and controlled for successful project deliveries. Results indicate that the evolutionary algorithms give better results (lower MMRE) as compared to the MMRE values of initial integrated effort estimate IE_{intial}. From table 3 it can be observed that among the evolutionary algorithms, PSO outperformed the other two algorithms (ABC and GLBPSO) with the lowest MMRE value of 0.131 in the shortest time. MMRE for GLBPSO was also 0.131, but it took more execution time than PSO.

Table 2: Parameter values for ABC, PSO and GLBPSO algorithms: Waterfall model

Weights	ABC	PSO	GLBPSO
w_{1}	0	0	0
w_{2}	0.168	0.552	0.579
w_{3}	0.596	1	0.686
w_{4}	0.826	1	1
w_{5}	0.954	1	1
w_{6}	0.685	1	0.729
w_{7}	0.755	1	1
w_{8}	0.33	0.631	0.525
w_{9}	0.884	1	1
w_{10}	0.695	0.818	0.935
w_{11}	0.334	0	0
w_{12}	0.624	0	0
w_{13}	0.796	0.79	0
w_{14}	0.891	0	1
w_{15}	0.302	1	0
w_{16}	0.551	0	0.742
w_{17}	0.713	0	0
w_{18}	0.437	0.733	0
w_{19}	0.301	1	0
w_{20}	0.549	0.736	0.678
w_{21}	0.917	0.159	0

Table 3: Waterfall Model experimental results



Waterfall Model	MMRE	SA	(Δ)	R^2	Time (seconds)
CoCoMo II	0.215	0.829	0.521	0.581	456.15
IE(Initial)	0.183	0.845	0.596	0.729	453.23
ABC	0.147	0.843	0.507	0.773	5.56×10^3
PSO	0.131	0.843	0.575	0.767	3.52×10^3
GLBPSO	0.131	0.843	0.545	0.791	5.4×10^3

In table 4, the integrated effort estimates obtained by the five approaches: CoCoMo II, $IE_{initial}$, ABC, PSO and GLBPSO are depicted project wise. These effort estimates are calculated by substituting the weight values listed in table 2 in equation (8). It can be noted that for most of the projects, integrated effort estimates for evolutionary algorithms are lower than their corresponding $IE_{initial}$ value given in column 4 of table 7. This indicates that the experts might have overestimated the alternative cost of the project cost factors that increased the project risk exposure. For projects (P1, P3, P5, P8, P16, P19, P33, P36, P40 and P41), the integrated effort for the evolutionary algorithms is higher than their corresponding $IE_{initial}$ value. The MRE for these projects were in the range: CoCoMo II (0%, 39%), $IE_{initial}$ (3%, 38%), ABC (1%, 37%), PSO (0%, 41%) and GLBPSO (0%, 36%). This indicates under estimation of risk exposure due to the project cost factors by the experts. Figure 1 shows the variation in MRE of projects for all the considered five estimation approaches. MRE for projects (P4, P7, P10, P11, P15, P17, P19, P20, P23, P24, P25, P26, P27, P30, P31, P33, P37, P39, P40, P41, P44 and P45) has reduced considerably with the use of evolutionary algorithms as compared to the MRE values obtained when using CoCoMo II or initial integrated effort estimates $IE_{initial}$.

Table 4: Estimated effort for Waterfall Model

Project Id	Actual Effort	CoCoMo II	$IE_{initial}$	ABC	PSO	GLBPSO
P1	1634	1440.801	1728.743	1904.43	2036.967	2024.036
P2	700	666.9536	713.9921	594.992	589.1942	564.2396
P3	3987	3852.068	4123.872	4309.477	4125.641	4343.761
P4	450	404.2893	498.517	441.5914	449.4359	442.9401
P5	2608	2702.005	2535.76	2652.95	2430.574	2594.073
P6	3567	3304.395	3661.615	3085.545	2944.122	2957.799
P7	2256	2048.078	2473.87	2150.134	2320.11	2295.657
P8	912	965.8243	952.08	969.2639	918.2394	978.07
P9	2879	2518.215	2959.183	2329.926	2673.046	2492.919
P10	2435	1922.343	2318.947	2048.018	2440.157	2285.411
P11	1456	1297.093	2202.691	1957.832	1884.466	1590.199
P12	2234	1480.343	2414.968	2262.714	2347.047	2421.157
P13	3200	2020.047	3080.992	2835.488	3174.499	2578.99
P14	3567	2339.519	4248.97	3801.656	3581.139	3753.184
P15	3678	2044.514	3905.35	3523.734	3309.443	3461.998
P16	2759	2756.707	2585.53	2705.774	2478.366	2645.365
P17	3015	4621.11	4901.983	4300.253	4093.611	4115.907
P18	3459	2897.416	3497.342	3053.948	3288.689	3253.642
P19	2435	1490.16	1981.087	2114.641	2201.621	2278.578
P20	859	1412.506	1496.216	1236.099	1208.915	1163.307
P21	3147	1821.516	2934.557	2384.43	1944.12	2307.894
P22	1987	1304.258	2239.509	1765.508	1465.091	1709.601
P23	4567	4632.826	5968.334	4640.557	4502.962	4047.402
P24	3629	3275.441	4936.89	4158.498	3676.23	4085.744
P25	2897	2937.634	4906.66	3734.724	3305.251	3671.437
P26	2453	1247.939	1496.43	1498.683	1644.663	1492.266
P27	3786	6163.266	5581.173	4626.056	4513.531	3955.972
P28	2687	2577.099	2510.667	2574.539	2437.885	2385.794

Project Id	Actual Effort	CoCoMo II	$IE_{initial}$	ABC	PSO	GLBPSO
P29	2937	2395.47	2887.848	2622.176	2534.778	2455.068
P30	2874	3730.238	3463.449	3229.824	2879.849	3033.284
P31	3384	3851.66	3346.509	3377.767	3371.97	3199.105
P32	3287	3480.278	3906.636	3347.397	3729.617	3474.419
P33	2845	2376.817	2534.226	2807.818	2936.245	2840.376
P34	3504	3542.165	3598.086	3204.293	3514.348	3175.856
P35	2134	2005.142	2034.214	1882.425	1901.643	1827.661
P36	2739	1746.16	1710.686	1734.193	1628.757	1740.316
P37	2469	2391.855	3657.088	3391.128	3802.217	3087.405
P38	3200	3963.386	3911.636	4026.358	3805.979	3724.716
P39	2489	2562.534	2590.451	2391.678	2407.534	2317.195
P40	1999	1411.736	1694.448	1865.935	1995.9	1983.034
P41	2598	1704.9	2264.622	2421.514	2521.187	2610.084
P42	2876	1369.664	2605.072	2319.319	2176.62	2281.717
P43	2309	2441.109	2374.307	2433.534	2304.895	2255.642
P44	1784	905.3366	1072.126	937.9648	1126.777	1054.362
P45	2792	3159.18	3502.639	2952.263	2818.442	2831.023

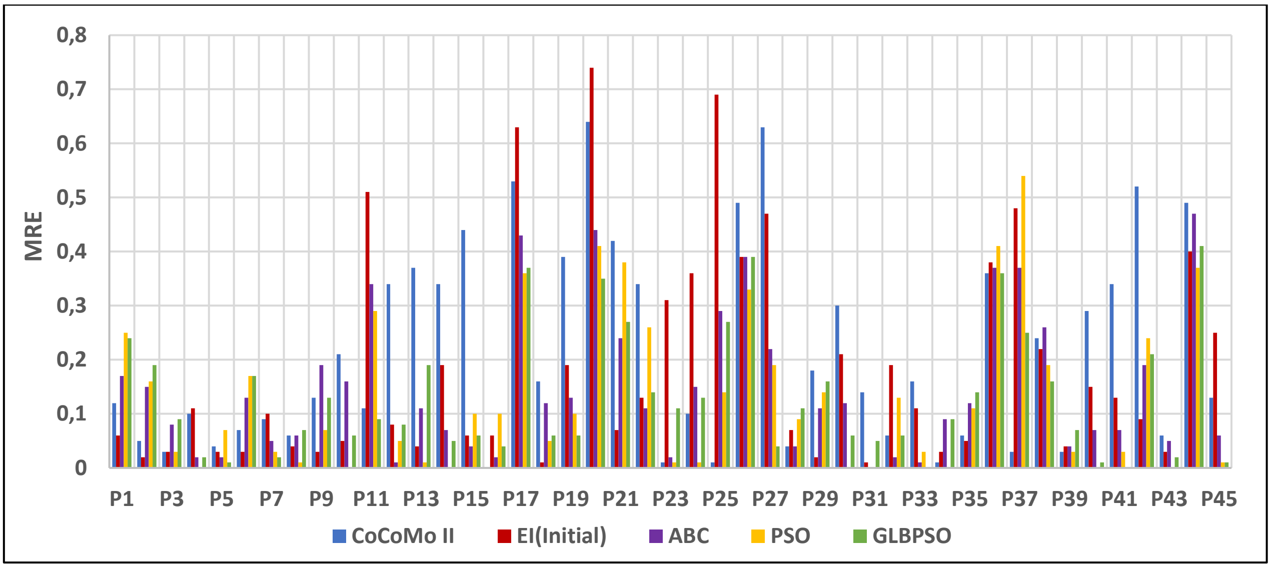
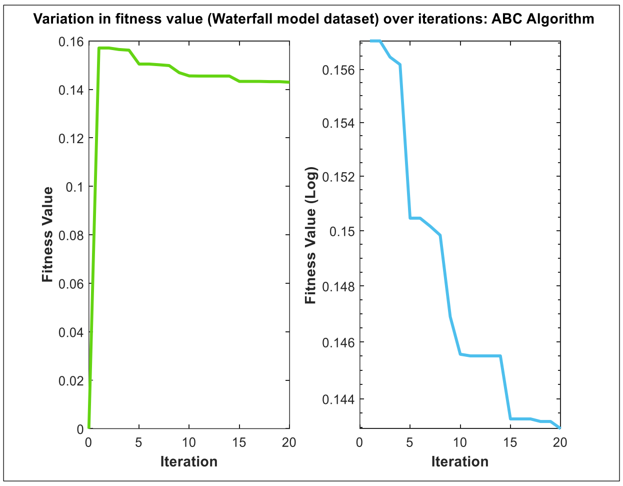


Figure 1: Variation of MRE in waterfall dataset



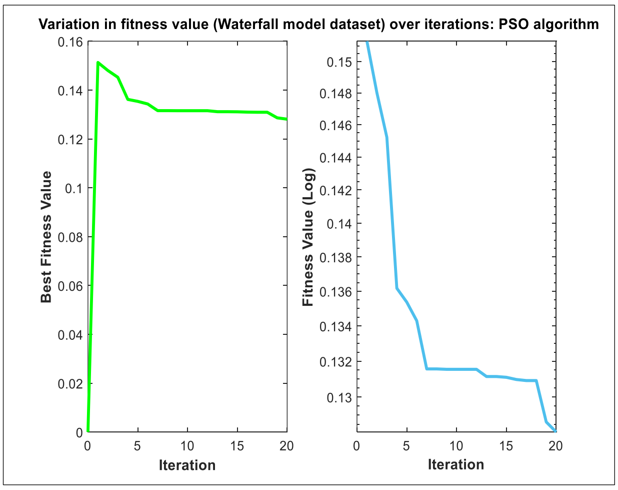


Figure 2: Variation in fitness value: ABC algorithm

Figure 3: Variation in fitness value: PSO algorithm

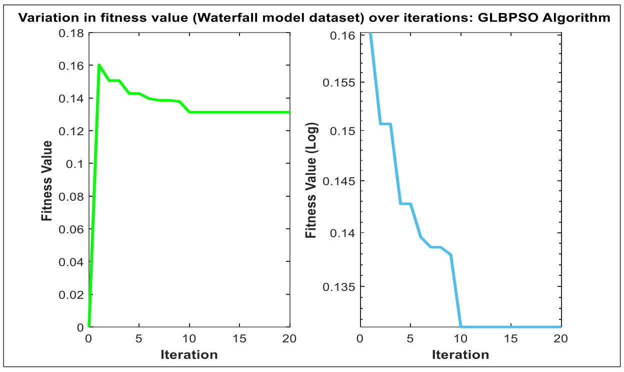


Figure 4: Variation in fitness value: GLBPSO algorithm

7.2 Results for agile model dataset

For the Agile dataset, the optimal weight values obtained for all the three algorithms: ABC, PSO and GLBPSO are listed in table 5. Results shown in table 6, indicate that the evolutionary algorithms gave better results (lower MMRE) as compared to the MMRE obtained for the initial integrated effort estimate (equation 3). It can also be observed that among the evolutionary algorithms, PSO outperformed the other two algorithms (ABC and GLBPSO) with the lowest MMRE value of 0.151 in the shortest time. MMRE for GLBPSO was also 0.151, but it took more execution time than the PSO.

Table 5: Parameter values for ABC, PSO and GLBPSO Algorithms for Agile model

Weights	**ABC**	**PSO**	**GLBPSO**
w1	0.9183	1	1
w2	0.0223	0	0
w3	0.603	0.783	0.781
w4	0	0	0
w5	0.0733	0	0
w6	1	1	1
w7	0.4551	0.3888	0.3936
w8	0.8715	1	1
w9	0.7779	1	1
w10	0.0417	0	0
w11	0.2197	0	0
w12	0.0697	0	0
w13	0.0396	0	0

The integrated effort estimates for the various approaches: Ziauddin, IE(Initial), ABC, PSO and GLBPSO are depicted in table 7. These effort estimates are calculated by substituting the weight values listed in table 5 in equation (11). For the Agile dataset, the MMRE (0.282) for IE(Initial) is close to the MMRE (0.288) of Ziauddin approach. The initial integrated effort estimates IE(Initial) for most of the projects are lower than the effort estimates using Ziauddin approach. The integrated effort estimates IE for all the evolutionary algorithms are higher than their IE(Initial) estimates. This indicates that the experts had assumed the cost of project cost factors optimistically. Use of evolutionary algorithms has considerably reduced this over optimism, as indicated by the MMRE values (ABC: 0.155, PSO, GLBPSO: 0.151). Figure 5 shows the variation of MRE obtained for projects for all the considered estimation approaches. MRE for projects (P3, P4, P5, P6, P9, P12, P15, P16, P17, P20, P21, P24, P26 and P30) has reduced considerably with the use of evolutionary algorithms as compared to the MRE values obtained when using Ziauddin approach or the initial integrated effort estimates.

Table 6: Agile model experimental results

Agile Model	**MMRE**	**SA**	**Effect Size (Δ)**	**R^2**	**Time (seconds)**
Ziauddin	0.288	1.85	0.603	0.018	356.36
IE(Initial)	0.282	2.14	0.713	0.102	347.56
ABC	0.155	2.34	0.756	0.113	9.32×10^3
PSO	**0.151**	**2.43**	**0.766**	**0.114**	$\mathbf{6.17 \times 10^3}$
GLBPSO	**0.151**	**2.44**	**0.765**	**0.113**	$\mathbf{1.19 \times 10^4}$

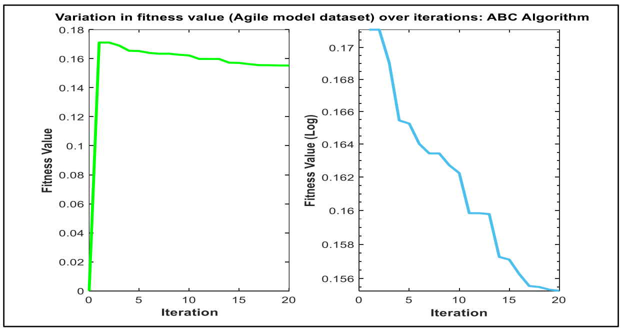


Figure 6: Variation in fitness value: ABC algorithm

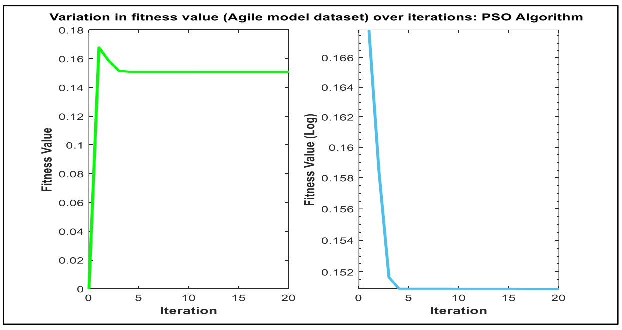


Figure 7: Variation in fitness value: PSO algorithm

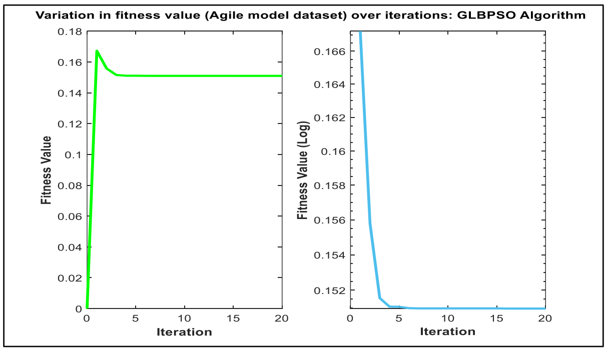


Figure 8: Variation in fitness value: GLBPSO algorithm

Table 7: Estimated effort for agile model

Project Id	Actual Effort	Ziauddin	IE (Initial)	ABC	PSO	GLBPSO
1	64	86.49231	83.60254	81.18652	82.4856	82.48179
2	76	117.1165	109.1453	109.792	110.9355	110.9644
3	81	145.9146	132.0753	125.3142	125.1441	125.1514
4	68	119.1528	107.4205	92.31646	93.95784	93.87866
5	66	57.66671	56.37771	62.58207	62.75219	62.77239
6	67	63.59279	60.88424	64.93346	65.75203	65.7274
7	67	43.43548	43.27209	45.72152	46.55416	46.55146
8	68	67.05171	52.21071	57.06398	58.15929	58.10352
9	56	56.27627	44.25048	50.32494	50.82362	50.8004
10	60	47.195	44.33398	46.6364	47.29976	47.30984
11	58	72.43631	59.08681	67.43594	68.02355	67.97869
12	62	48.89734	45.28741	53.88648	54.93635	54.95424
13	65	48.81696	47.96385	53.45084	53.79691	53.81199
14	63	60.65094	52.03628	50.69616	51.91528	51.87922
15	65	56.06237	55.76618	63.89089	65.00141	65.01872
16	66	77.1008	45.07614	61.89129	64.40236	64.34528
17	65	70.89737	58.16727	67.64816	68.42956	68.44052
18	62	57.01432	55.20099	56.30309	56.1034	56.12013
19	76	57.59758	57.12901	61.91986	62.78898	62.80839
20	75	75.05299	65.01085	69.15184	70.20775	70.20775
21	73	70.55185	64.72608	68.51258	69.79861	69.78006
22	72	61.62484	53.08437	58.69321	58.78924	58.80578
23	73	70.14982	67.56121	81.80111	83.00832	83.00832
24	74	89.07373	76.97409	74.24416	75.22673	75.20611
25	60	49.52664	42.81767	49.61795	50.55862	50.54135
26	65	77.78041	59.7399	65.04337	65.00297	64.99734
27	62	58.08273	42.89178	49.42383	49.86158	49.86158
28	55	47.45795	46.40748	47.02344	47.1733	47.17914
29	60	66.81642	61.06547	63.36946	63.95546	63.93309
30	59	64.98048	44.34687	55.10337	56.39194	56.37387

7.3 Revisiting the research questions

RQ1: *Does the accuracy of effort estimate of the project improve by adding the cost of risk exposure to the initial estimated effort of the project?*

Results as discussed in sections 7.1 and 7.2, show that the integrated effort estimates have lower values of MMRE and higher values of SA, effect size and R^2 than the corresponding initial effort estimates, for both the datasets. Thus, it can be concluded that the integrated effort estimates are more accurate, reliable, and comprehensive than the initial effort estimates.

RQ2: *What is the impact of bias on risk exposure of the project?*

Results in sections 7.1 and 7.2, show that the MMRE of the software effort estimate is reduced by using evolutionary algorithms ABC, PSO and GLBPSO for both the datasets. In the process, we also obtain the optimum weight value, w_i, corresponding to the i^{th} project cost factor, which is the optimum value of $p_{i,alter}$. These optimal values of the $w_i's$ reflect the unbiased values of $(p_{i,alter})'s$.

RQ3: *Can project cost factors be ranked with respect to their risk exposure?*

High value of w_i implies that the contribution of the i^{th}project cost factor is also high in the risk exposure of the project. So, the values of w_i's are good indicators of project cost factors which have high risk exposure.

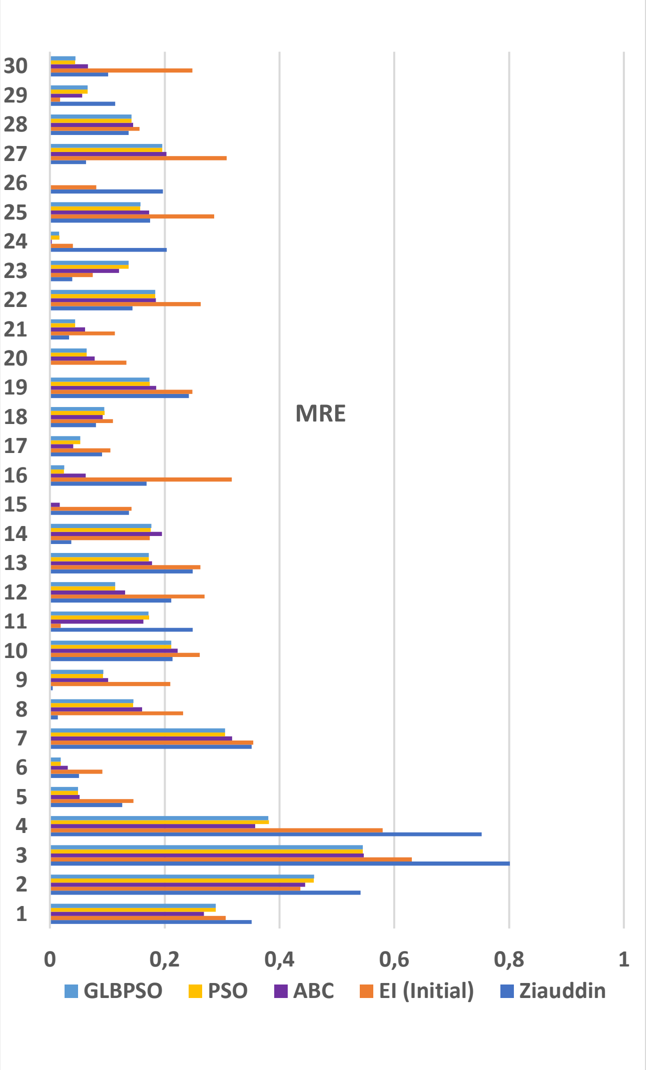


Figure 5: Variation of MRE in agile dataset

8 Threats to validity

This section discusses threats to validity of the proposed model.

External validity

External validity [42] is concerned with generalization of the results obtained. Threats to external validity are conditions that limit the ability to generalize the results of the proposed experiment to other effort estimation models [43-46].

For this research, the datasets have been collected from a single organization. Although the organization is a large IT consultancy firm working on projects in varied domains, the proposed approach could be further validated by experimenting on data from different organizations. To test the efficacy of evolutionary algorithms, large datasets are ideal. The results in this paper are based on datasets with 30 to 45 projects.

9 Conclusion

This paper has introduced a novel approach for integrating the impact of risk exposure into the effort estimate of a software project. This impact is determined using the weights associated with the risk exposure due to each project cost factor. These weights are then optimized using evolutionary approaches like ABC, PSO and GLBPSO. Experimental results show that the PSO and GLBPSO algorithms gave more accurate effort estimates for both waterfall and agile projects, but GLBPSO took more time. The approach essentially reduces the bias due to the probabilities which were associated with the impact of risk exposure on the effort estimates of the projects. Software effort estimation for projects now, will, not have to rely solely on the expert judgment for assessing the probable impact of the risk exposure due to project cost factors.

The project factors can be ranked based upon the associated optimal weight values. Software Project managers can prepare and plan for risk management and development of the project effectively using the ranking obtained. Cost factors with higher weight values will need to be mitigated and controlled earlier than the cost factors with lower weight values.

10 Future directions

In the manuscript, tables 4 & 7 list the effort estimates calculated by using the optimum weight values obtained by applying ABC, PSO and GLBPSO on waterfall and agile model datasets respectively. To validate the obtained results, tables 5 and 6, then compare the calculated effort estimates based on MMRE, SA, Effect size and R^2. The results obtained confirm that the risk integrated effort estimation accuracy improves with the application of evolutionary algorithms such as ABC, PSO and GLBPSO. The proposed risk integrated approach can further be validated through additional case study / company data.

The proposed risk integrated effort estimation approach can be applied to other benchmark effort estimation models such as Use Case Point [47], Function Point [48], and Analogy based estimation [49] for Waterfall projects. Poker [50], T-shirt sizing and Three point estimation [51] for Agile projects. To enable the comparison, cost factor data for the suggested benchmark models will have to be collected / generated.

To further investigate the impact of evolutionary algorithms on weight values associated with cost factors, other available evolutionary algorithms such as firefly, ant colony optimization, cuckoo search and whale optimization could be used, and results compared with the results obtained in this research. The weight values can also be optimized using artificial intelligence techniques like neural networks, convolutional neural networks, and deep learning techniques.

11 Declarations

Funding

Not Applicable

Availability of data and material

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Code availability

Not Applicable

References

[1] S. Hastie and S. Wojewoda, "Standish group 2015 chaos report-q&a with jennifer lynch," *Retrieved,* vol. 1, p. 2016, 2015.

[2] D. Eck, B. Brundick, T. Fettig, J. Dechoretz, and J. Ugljesa, "Parametric estimating handbook," *The International Society of Parametric Analysis (ISPA),* 2009.

[3] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobylinski, "An effective approach for software project effort and duration estimation with machine learning algorithms," *Journal of Systems and Software,* vol. 137, pp. 184-196, 2018. https://doi.org/10.1016/j.jss.2017.11.066

[4] O. Morgenshtern, T. Raz, and D. Dvir, "Factors affecting duration and effort estimation errors in software development projects," *Information and Software Technology,* vol. 49, pp. 827-837, 2007. https://doi.org/10.1016/j.infsof.2006.09.006

[5] K. Kansala, "Integrating risk assessment with cost estimation," *IEEE software,* vol. 14, pp. 61-67, 1997. https://doi.org/10.1109/52.589236

[6] B. W. Boehm, "Software risk management: principles and practices," *IEEE software,* vol. 8, pp. 32-41, 1991.
https://doi.org/10.1109/52.62930

[7] M. Jørgensen, "Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates," *Information and Software Technology,* vol. 52, pp. 506-516, 2010. https://doi.org/10.1016/j.infsof.2009.12.002

[8] P. Singal, P. Sharma, and A. C. Kumari, "Integrating software effort estimation with risk management,"

International Journal of System Assurance Engineering and Management, pp. 1-16, 2022. https://doi.org/10.1007/s13198-022-01652-y

[9] B. Kitchenham and S. Linkman, "Estimates, uncertainty, and risk," *IEEE Software,* vol. 14, pp. 69-74, 1997. https://doi.org/10.1109/52.589239

[10] B. Boehm, "Software risk management," in *European Software Engineering Conference*, 1989, pp. 1-19. https://doi.org/10.1007/3-540-51635-2_29

[11] R. Fairley, "Risk management for software projects," *IEEE software*, vol. 11, pp. 57-67, 1994. https://doi.org/10.1109/52.281716

[12] R. J. Madachy, "Heuristic risk assessment using cost factors," *IEEE software,* vol. 14, pp. 51-59, 1997. https://doi.org/10.1109/52.589234

[13] L. C. Briand, K. El Emam, and F. Bomarius, "COBRA: a hybrid method for software cost estimation, benchmarking, and risk assessment," in *Proceedings of the 20th international conference on Software engineering*, 1998, pp. 390-399. https://doi.org/10.1109/ICSE.1998.671392

[14] K. Jantzen, "Estimating the effects of project risks in software development projects," 2006.

[15] S.-J. Huang, C.-Y. Lin, and N.-H. Chiu, "Fuzzy decision tree approach for embedding risk assessment information into software cost estimation model," *Journal of information science and engineering,* vol. 22, pp. 297-313, 2006.

[16] E. Manalif, "Fuzzy Expert-COCOMO risk assessment and effort contingency model in software project management," 2013.

[17] W. Aslam, F. Ijaz, M. I. U. Lali, and W. Mehmood, "Risk Aware and Quality Enriched Effort Estimation for Mobile Applications in Distributed Agile Software Development," *J. Inf. Sci. Eng.*, vol. 33, pp. 1481-1500, 2017. https://doi.org/10.6688/JISE.2017.33.6.6

[18] S. El Koutbi and A. Idri, "Software Effort Estimation Risk Management over Projects Portfolio," *Comput. Inf. Sci.*, vol. 11, pp. 45-76, 2018. https://doi.org/10.5539/cis.v11n4p45

[19] N. Ramakrishnan, H. Girijamma, and K. Balachandran, "Enhanced Process Model and Analysis of Risk Integration in Software effort estimation," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2019, pp. 419-422. https://doi.org/10.1109/ICSSIT46314.2019.8987841

[20] M. Kataev, L. Bulysheva, L. Xu, Y. Ekhlakov, N. Permyakova, and V. Jovanovic, "Fuzzy model estimation of the risk factors impact on the target of promotion of the software product," *Enterprise Information Systems,* vol. 14, pp. 797-811, 2020. https://doi.org/10.1080/17517575.2020.1713407

[21] M. Azzeh, A. B. Nassif, and S. Banitaan, "Comparative analysis of soft computing techniques for predicting software effort based use case points," *IET Software,* vol. 12, pp. 19-29, 2018. https://doi.org/10.1049/iet-sen.2016.0322

[22] T. T. Khuat and M. H. Le, "A novel hybrid abc-pso algorithm for effort estimation of software projects using agile methodologies," *Journal of Intelligent Systems*, vol. 27, pp. 489-506, 2018. https://doi.org/10.1515/jisys-2016-0294

[23] D. Novitasari, I. Cholissodin, and W. F. Mahmudy, "Hybridizing PSO with SA for optimizing SVR applied to software effort estimation," *Telkomnika (Telecommunication Computing Electronics and Control),* vol. 14, pp. 245-253, 2016. http://doi.org/10.12928/telkomnika.v14i1.2812

[24] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization*, vol. 39, pp. 459-471, 2007. http://doi.org/0.1007/s10898-007-9149-x

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, 1995, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[26] R. R. Chhikara, P. Sharma, and L. Singh, "A hybrid feature selection approach based on improved PSO and filter approaches for image steganalysis," *International Journal of Machine Learning and Cybernetics,* vol. 7, pp. 1195-1206, 2016. https://doi.org/10.1007/s13042-015-0448-0

[27] T. Gilb, "Evolutionary Delivery versus the" waterfall model"," *ACM sigsoft software engineering notes*, vol. 10, pp. 49-61, 1985. https://doi.org/10.1145/1012483.1012490

[28] R. C. Martin, M. Martin, and M. Martin, *Agile principles, patterns, and practices in C#*: Prentice Hall, 2007.

[29] B. Boehm, C. Abts, B. Clark, and S. Devnani-Chulani, "COCOMO II model definition manual," *The University of Southern California,* 1997.

[30] S. K. T. Ziauddin and S. Zia, "An effort estimation model for agile software development," *Advances in computer science and its applications (ACSA),* vol. 2, pp. 314-324, 2012.

[31] S. Dalal, N. Dahiya, and V. Jaglan, "Efficient Tuning of COCOMO Model Cost Drivers Through Generalized Reduced Gradient (GRG) Nonlinear Optimization with Best-Fit Analysis," in *Progress in Advanced Computing and Intelligent Engineering*, ed: Springer, 2018, pp. 347-354. https://doi.org/10.1007/978-981-10-6872-0_32

[32] A. W. M. M. Parvez, "Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development," in *Information Technology and Electrical Engineering (ICITEE), 2013 International Conference on*, 2013, pp. 113-118. https://doi.org/10.1109/ICITEED.2013.6676222

[33] T. Urbanek, Z. Prokopova, R. Silhavy, and V. Vesela, "Prediction accuracy measurements as a fitness function for software effort estimation," *SpringerPlus,* vol. 4, pp. 1-17, 2015. https://doi.org/10.1186/s40064-015-1555-9

[34] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit, "A simulation study of the model evaluation criterion MMRE," *IEEE transactions on software engineering,* vol. 29, pp. 985-995, 2003. https://doi.org/10.1109/TSE.2003.1245300

[35] B. A. Kitchenham, L. M. Pickard, S. G. MacDonell, and M. J. Shepperd, "What accuracy statistics really measure," *IEE Proceedings-Software,* vol. 148, pp. 81-85, 2001. https://doi.org/10.1049/ip-sen:20010506

[36] M. Korte and D. Port, "Confidence in software cost estimation results based on MMRE and PRED," in *Proceedings of the 4th international workshop on Predictor models in software engineering*, 2008, pp. 63-70. https://doi.org/10.1145/1370788.1370804

[37] D. Port and M. Korte, "Comparative studies of the model evaluation criterions mmre and pred in software cost estimation research," in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, 2008, pp. 51-60. https://doi.org/10.1145/1414004.1414015

[38] A. Idri, I. Abnane, and A. Abran, "Evaluating Pred (p) and standardized accuracy criteria in software development effort estimation," *Journal of Software: Evolution and Process,* vol. 30, p. e1925, 2018. https://doi.org/10.1002/smr.1925

[39] E. Stensrud, T. Foss, B. Kitchenham, and I. Myrtveit, "A further empirical investigation of the relationship between MRE and project size," *Empirical software engineering,* vol. 8, pp. 139-161, 2003. https://doi.org/10.1023/A:1023010612345

[40] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Computational Intelligence and neuroscience,* vol. 2019, 2019. https://doi.org/10.1155/2019/8367214

[41] N. J. Nagelkerke, "A note on a general definition of the coefficient of determination," *Biometrika,* vol. 78, pp. 691-692, 1991.

[42] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*: Springer Science & Business Media, 2012.

[43] A. Idri, F. azzahra Amazal, and A. Abran, "Analogy-based software development effort estimation: A systematic mapping and review," *Information and Software Technology,* vol. 58, pp. 206-230, 2015. https://doi.org/10.1016/j.infsof.2014.07.013

[44] J. Popovic, D. Bojic, and N. Korolija, "Analysis of task effort estimation accuracy based on use case point size," *IET Software,* vol. 9, pp. 166-173, 2015. https://doi.org/10.1049/iet-sen.2014.0254

[45] I. Hussain, L. Kosseim, and O. Ormandjieva, "Approximation of COSMIC functional size to support early effort estimation in Agile," *Data & Knowledge Engineering,* vol. 85, pp. 2-14, 2013. https://doi.org/10.1016/j.datak.2012.06.005

[46] M. Usman, "Improving Expert Estimation of Software Development Effort in Agile Contexts," Blekinge Tekniska Högskola, 2018.

[47] M. R. Braz and S. R. Vergilio, "Software effort estimation based on use cases," in *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, 2006, pp. 221-228. https://doi.org/10.1109/COMPSAC.2006.77

[48] A. Hira and B. Boehm, "COSMIC Function Points Evaluation for Software Maintenance," in *Proceedings of the 11th Innovations in Software Engineering Conference*, 2018, p. 4. https://doi.org/10.1145/3172871.3172874

[49] M. Azzeh and A. B. Nassif, "Analogy-based effort estimation: a new method to discover set of analogies from dataset characteristics," *IET Software,* vol. 9, pp. 39-50, 2015. https://doi.org/10.1049/iet-sen.2013.0165

[50] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," *Journal of Systems and Software,* vol. 85, pp. 2086-2095, 2012. https://doi.org/10.1016/j.jss.2012.04.005

[51] R. K. Mallidi and M. Sharma, "Study on agile story point estimation techniques and challenges," *Int. J. Comput. Appl,* vol. 174, pp. 9-14, 2021.

O-AIRS: Optimized Artificial Immune Recognition System

Badr Merad[1,2*], Khaled Belkadi[3]
[1] Laboratory of Electrical Engineering and Materials (LEEM), Oran, Algeria
[2] Higher School in Electrical and Energetic Engineering of Oran (ESGEEO), Oran, Algeria
[3] Faculty of mathematics and computer sciences, University of Sciences and Technology of Oran Mohamed Boudiaf, Oran, Algeria
Email: merad_badr@esgee-oran.dz, belkadi1999@yahoo.com
*Corresponding author

Keywords: AIRS, Artificial Immune Systems, Artificial Intelligence, Classification, Clonal Selection, Supervised Learning

Received: February 15, 2023

Artificial Immune Recognition System (AIRS) offers a promising meta-heuristic approach inspired by the human immune system for classification tasks. However, limitations such as reliance on single-antigen activation and retention of untested memory cells can lead to inaccuracies. This paper proposes the Optimized Artificial Immune Recognition System (O-AIRS) to mitigate these issues. O-AIRS leverages Homogeneous Antigen Groups (HAGs) for refined memory cell activation, ensuring a precise threat response. Furthermore, O-AIRS incorporates a robust maturity mechanism to retain only validated memory cells, enhancing classification accuracy. The effectiveness of O-AIRS was assessed using established medical datasets: Liver Disorders (LD) and Haberman Surgery Survival (HSS). Experimental evaluation on both LD and HSS datasets establishes O-AIRS's superiority over AIRS and AIRS2 across various performance metrics. Notably, O-AIRS achieves this enhanced performance while utilizing approximately 50% fewer memory cells during classification due to its optimized activation mechanism. Importantly, O-AIRS guarantees the maturity of all memory cells, ensuring effective threat recognition.

Povzetek: Predstavljen je sistem za umetno prepoznavanje imunosti (O-AIRS) izboljšuje natančnost klasifikacije z uporabo homogenih antigenskih skupin (HAG) in mehanizma zrelosti za shranjevanje preverjenih spominskih celic, kar je dokazano učinkovitejše od AIRS in AIRS2 na medicinskih podatkovnih nizih, hkrati pa uporablja približno 50 % manj spominskih celic.

Povzetek: Predstavljen je sistem za umetno prepoznavanje imunosti (O-AIRS), ki izboljšuje klasifikacije z uporabo homogenih antigenskih skupin (HAG) in mehanizma zrelosti za shranjevanje preverjenih spominskih celic, kar je dokazano učinkovitejše od AIRS in AIRS2 na medicinskih podatkovnih nizih.

1 Introduction

The ability to learn and adapt is fundamental to human intelligence. It allows individuals to improve through experience, remember past decisions, and make better choices in similar situations. This concept underpins much of modern Artificial Intelligence (AI) research, which focuses on developing systems capable of learning and decision-making akin to human cognition.

Inspired by biological systems such as the human brain, neurons, and genetic processes, researchers develop bio-inspired approaches to AI. These methods seek to replicate the efficiency and adaptability observed in natural systems, paving the way for advancements in machine learning algorithms.

Among these bio-inspired approaches, Artificial Immune Systems (AIS) stand out for their emulation of the human immune system's ability to recognize and respond to threats. Originally conceived in the 1950s, AIS models have evolved to tackle complex computational challenges, including pattern recognition, anomaly detection, and optimization tasks.

Within the realm of AIS, the Artificial Immune Recognition System (AIRS) [1] and its successor, AIRS2 [2], have garnered significant attention for their effectiveness in supervised learning tasks. These algorithms are known for their good classification capabilities and strong capacities to support decision makers and resolve real-world issues [3].

While AIRS and AIRS2 have demonstrated success, their effectiveness can be hampered by inherent limitations. These limitations include dependence on single antigens for memory cell activation and the potential use of untested memory cells during classification.

This paper addresses these limitations by introducing a novel AIS algorithm: the Optimized Artificial Immune Recognition System (O-AIRS). O-AIRS leverages innovative methodologies like Homogeneous Antigen Groups (HAGs) and a refined memory cell activation mechanism to overcome these shortcomings. HAGs enhance pattern recognition, and the refined activation mechanism ensures robust and reliable classifications. These advancements aim to improve classification accuracy and computational efficiency for diverse datasets.

To comprehensively evaluate O-AIRS's effectiveness, the authors conducted a series of rigorous experiments utilizing established benchmark datasets. These datasets, Liver Disorders (LD) and Haberman Surgery Survival (HSS), represent real-world scenarios commonly encountered in the medical domain. By employing a diverse range of metrics to assess O-AIRS's performance, the experiments provide a robust and generalizable understanding of its capabilities.

This paper is structured as follows: Section 2 discusses existing research relevant to Artificial Immune Systems. Section 3 provides a detailed explanation of the Artificial Immune Recognition System (AIRS). Section 4 explores the advancements made in AIRS2. Section 5 highlights the key innovations introduced in O-AIRS compared to AIRS/AIRS2. Section 6 presents the experimental study, including the use of the Haberman Surgery Survival (HSS) and Liver Disorder (LD) datasets, and summarizes the overall findings. Finally, Section 7 concludes the paper by summarizing the research findings and outlining future research directions.

2 Related works

Artificial Immune Systems (AIS) have been extensively researched, resulting in a variety of algorithms. These algorithms mainly fall into four categories: negative selection, immune network, danger theory, and clonal selection [4].

Negative selection algorithms, originating from the seminal work of Forrest et al. [5], emulate how the immune system identifies and eliminates antibodies that mistakenly recognize 'self' components as antigens. This process is crucial in data security contexts, where 'self' corresponds to the data to be protected and 'non-self' denotes potentially harmful data. Subsequent studies, such as [6], [7], and [8], have further refined and extended these mechanisms, leading to enhanced efficacy of self-nonself discrimination.

The concept of an immune network, a self-regulating system that distinguishes "self" from "non-self" without direct contact with antigens, was introduced by Jerne in 1974 [9]. This foundational idea laid the groundwork for Artificial Immune Networks (AINs). Building on this work, Varela and Coutinho [10] developed a second-generation model focused on enhancing adaptability and response dynamics within the network and De Castro and Von Zuben [11] proposed aiNet. These networks offer advantages similar to the human immune system, such as the ability to adapt to new threats. Recent research like [12], [13] and [14] continues to explore and optimize AIN frameworks for diverse applications, including pattern recognition and optimization problems.

The danger theory, proposed by Matzinger [15], suggests that immune responses are initiated not only through direct interactions between antibodies and antigens, but also by signals released during cellular damage or stress. This concept has played a crucial role in the evolution of Intrusion Detection Systems (IDSs), enabling the detection of potential threats while allowing harmless antibodies and antigens to coexist in the absence of danger signals. This approach has been successfully implemented and further developed in practical Intrusion Detection systems, exemplified by notable works such as those referenced in [16], [17] and [18].

The clonal selection principle, initially formulated by Burnet and refined through computational models, describes how antibodies proliferate upon encountering antigens, followed by mutations to enhance affinity [19]. Seminal contributions in this area include CLONALG (CLONal selection ALGorithm) [20] and AIRS (Artificial Immune Recognition System) [1]. CLONALG introduced fundamental concepts of clonal selection in computational models [21], laying the groundwork for subsequent advancements. Ongoing research [22], [2], [23] and [24] continues to explore and refine clonal selection mechanisms, with the aim of enhancing adaptability and efficiency across various problem domains. AIRS [1] and its successor AIRS2 [2] have gained recognition for their robust classification capabilities and practical application in decision support systems [3]. Table 1 provides a comprehensive overview of the key references discussed in this section.

Table 1: Summary of relevant literature.

	Algorithm	Dataset	Application area	Performance metric	Strengths	Limitations	Remarks
Negative Selection	Self-Nonself Discrimination [5]	Experimental data	Virus detection	· Probability of detection · Computational cost	Protects anti bodies from modification to conform to altered self	High computational cost for generating the initial "self" repertoire	First negative selection algorithm

	Improved Negative Selection Algorithm (Improved-NSA) [6]	Sensor data: three-tank system	Fault detection	· Coverage Rate · Overlap Rate · Detection Rate · False Alarm Rate	· Automatic adjustment of detector radius for optimal coverage · Reduces overlap between detectors	· Requires predefined "self" data for training · May not be suitable for complex anomaly patterns	Custom datasets for three-tank system (not public)
	Antigen Density Clustering - Negative Selection Algorithm (ADC-NSA) [7]	· Breast Cancer Wisconsin (BCW) · Knowledge Discovery and Data Mining Cup 99 (KDD-Cup99)	Anomaly detection: · Medical diagnosis · Network intrusion detection · Spam detection	· Detection Rate · False Positive Rate	· Addresses uneven antigen distribution · Improves detection efficiency · Reduces randomness in detector generation	· Requires tuning cutoff distance for clustering · Needs further research on identifying "loopholes" (undetected data)	-
	Improved Negative Selection Algorithm (INSA) [8]	Normal state sample library of distribution network dataset	High resistance fault identification in distribution network	· Detection Rate · Classification · Accuracy	Requires small number of samples for training	Vulnerable to black holes (cannot classify when there are many types of abnormal states)	Custom dataset (not public)
Artificial Immune Networks	Artificial Immune Networks (AINet) [11]	Unlabeled numerical datasets	Data clustering and filtering	· Compression Rate · Classification Accuracy · Reduction of Redundancy	· Reduces data redundancy · Identifies groups and subgroups in the data · Determines the number of clusters and their structure · Offers good compression rates	· High number of user-defined parameters · Computationally expensive ($O(p^3)$) · Sensitive to suppression threshold	50-sample classification (5 classes) and the two-donut problem
	Adaptive Artificial Immune Networks [12]	· KDD'99 · CAIDA'07 · CAIDA'08	Network security, specifically Denial-of-Service (DoS) flooding attack detection and mitigation	· True Positive Rate · False Positive Rate · Hit Rate · Entropy	· Adapts thresholds for anomaly detection · Implements quarantine zones to isolate threats	· Requires careful tuning of parameters for optimal performance · May have computational overhead	Other data generated for this study by the DDoSIM tool were used in simulations
	COVID-opt-aiNet [14]	· COVID-19 CT dataset · COVID-19 Radiography dataset · Chest X-ray dataset	COVID-19 detection	· Precision · Recall · F1-Score · Accuracy	· Improved accuracy compared to standalone DL/ML methods · Reduced training time	· Requires large datasets for training · Performance might be dependent on specific datasets	Hybrid approach using AINet with DL/ML techniques

Danger Theory	Deep Dendritic Cell Algorithm (DeepDCA) [16]	IoT-Bot dataset	IoT intrusion detection	· Accuracy · Precision · Recall · F1-Score · False Alarm Rate	· Detects various IoT attacks (DoS, DDoS, information gathering, theft) · High accuracy (over 98.73%) · Low False Positive Rate	Requires large datasets for training	-
	Multi-Level Intrusion Detection System Based on Immune Theory [17]	Custom data generated by cooja simulator	Intrusion detection in Wireless Sensor Networks (WSNs)	· Detection Rate · Packet Overhead · Energy Overhead	Distributed and lightweight approach	Detection probability varies for different attacks.	-
	Danger Theory - Dendritic Cell Algorithm (DT-DCA) [18]	Custom data generated by simulation and from real Wireless Sensor Network (WSN)	Wireless Sensor Networks (WSNs)	· True Positive (TP) · False Negative (FN) · True Negative (TN) · False Positive (FP)	· Low FP Rate · Low Energy Consumption	Lower TP rates compared to some other methods	Simulated and real-world WSN platform implementation
Clonal Selection	CLONALG [20]	Custom character set 30-city instance of the travelling salesman problem	· Pattern recognition · Optimization	· Accuracy (Pattern Recognition) · Fitness Function Value (Optimization)	· Efficient for multimodal problems · Good convergence speed	· Sensitive to parameters · May require significant memory for large datasets	-
	Improved Clonal Selection Algorithm with K-Nearest Neighbor (ICSAT-KNN) [22]	· Brain Tumor dataset · Leukemia dataset · Prostate Tumor dataset	Cancer classification	Classification Accuracy	Competitive accuracy (96.36% average)	Relies on a small subset of genes (16% average) Requires parameter tuning (Gsize, K) for optimal performance	-
	CLONALG-M [23]	Custom generated data	Wireless Sensor Networks (WSNs)	· First Node Dies (FNA) · Half of the Nodes Alive (HNA) · Total Remaining Energy (TRE)	Improved performance of fuzzy clustering algorithms	· Computationally expensive · Only approximates optimal solution	-

	Artificial Immune Recognition System (AIRS) [1]	· Two datasets of points in 10x 10 space · The Fisher Iris dataset · Pima diabetes dataset · The Sonar dataset · Ionosphere dataset	Classification	· Accuracy · Final memory cells number	· Good classification capabilities · Strong capacities to support decision makers and resolve real-world issues	· Memory cells activation mechanism solicited by a single antigen · Retention of memory cells even if they have never been into contact with antigens	· The first two datasets are randomly generated · Data space in the first dataset is linearly separable · In the second dataset data space is not linearly separable
	Artificial Immune Recognition System 2 (AIRS2) [2]	· The Fisher Iris dataset · Pima diabetes dataset · Ionosphere dataset · The Sonar dataset	· Signal processing · Medical diagnosis · Biology	· Accuracy · Final memory cells number	· Good classification capabilities · Refinement of AIRS	· Memory cells activation mechanism solicited by a single antigen · Retention of memory cells even if they have never been into contact with antigens	-
	Constant Length Multi-objective Clonal Selection Optimization Algorithm (CL-MCSOA) [24]	· Yeast Sporulation · Yeast Cell Cycle · Arabidopsis Thaliana · Human Fibroblasts Serum · Rat CNS · Colon Tumor	Gene expression clustering	· Silhouette width index · Deviation · Connectivity · Dunn-index · Execution time	· Multi-objective · Robust · Faster convergence	· Complex design · User needs to define final clustering	All 6 datasets used to evaluate the algorithm are publicly available

In summary, previous studies, such as those on AIRS [1] and AIRS2 [2], have extensively explored immune-inspired algorithms. However, optimizing their classification capabilities and practical applications remains challenging. This study introduces O-AIRS, an Optimized AIRS algorithm, which refines the clonal selection metaphor to enhance classification performance.

3 Artificial immune recognition system (AIRS)

"The artificial immune systems are computational models inspired by the biological immune system" [25]. This section presents AIRS, a widely used AIS algorithm that describes how memory cells recognize antigens [26].

AIRS is divided into two phases: a learning phase for generating memory cells and a classification phase that utilizes these cells.

3.1 The learning phase

3.1.1 Initialization step

The first step of initialization involves normalizing the data such that the Euclidean distance between any two vectors (antigens) falls within the interval [0, 1]. Following normalization, the initial set of memory cells

(MC) and the set (P) of Artificial Recognition Balls (ARBs), which represent lymphocytes with attributes of vector, resource, and class, are randomly created from the training dataset.

Subsequently, the affinity_threshold parameter is computed to represent the average affinity between all pairs of antigens using formula 1.

$$Affinity\_Threshold = \frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n} affinity(ag_i,ag_j)}{\frac{n(n-1)}{2}} \quad (1)$$

Where:

- n is the number of antigens.
- ag_i, ag_j are the i^th and j^th antigens.
- affinity(X,Y) represents the Euclidean distance between two vectors X and Y.

3.1.2 Learning by antigens

After initialization is complete, each element of the training set (each antigen) is presented to the AIRS algorithm to learn from its characteristics. For each antigen ag, a sequence of steps is repeated:

1. Selection of the mc_{match} cell: this is the closest cell to the antigen ag in the shape space. mc_{match} and ag belong to the same class (see formula 2).

$$mc_{match} = argmax_{mc \in MC_{ag.c}} stimulation\ (ag,mc) \quad (2)$$

Where:

- $stimulation\ (x,y) = \begin{cases} 1 - affinity(x,y)\ if\ x.c \equiv y.c \\ \quad Affinity\ (x,y)\ else \end{cases}$
- MC_c represent MC set cells of the class c.
- ag.c is the class of ag.

The number of clones generated from mc_{match} depends on its affinity with the antigen (ag) (see formula 3)

$$nb\_clones = hyper\_clonal\_rate*clonal\_rate*stim \quad (3)$$

Where:

- nb_clones is the number of clones produced.
- hyper_clonal_rate is the maximum cloning rate.
- clonal_rate is the average cloning rate.
- stim = stimulation(mc_{match},ag).

2. Mutation of ARBs: each ARB generated by mc_{match} goes through a mutation function detailed in [27]. If the result is positive (mutation performed) the ARB is added to the set (P).

3. Calculation of resources for ARBs: resources are calculated for each element of (P), the cells closest to ag will experience higher stress and consequently receive more resources.

4. Clonage and mutation of ARBs: ARBs in this step are selected based on their affinity with the antigen ag.

5. Stimulation_threshold verification: the algorithm verifies that while the average stimulation of each group of ARBs of the same class is lower than a given value of "stimulation_threshold," it resumes from step (3).

6. Choice of the candidate cell: the candidate cell is selected from the set (P) based on its closest similarity to the antigen ag. This candidate cell is added to the set of memory cells (MC) only if its stimulation value with ag is higher than the stimulation value of mc_{match}.

The above steps are repeated for each antigen until the entire training set has been exhausted.

3.2 The classification phase

After the learning phase, the memory cells are ready to be used in the classification phase, which employs the K-Nearest Neighbors (KNN) algorithm. Each memory cell is presented to the data vector for stimulation. The classification system determines the vector's class based on the classes of its k-nearest memory cells.

4 AIRS2

AIRS2 is a refinements of AIRS developed in [2]. The changes made are:

- The initialization of the set of ARBs (P) is no longer necessary.
- Mutations now only concern data vectors and not classes.
- For the clonal selection and the criterion for stopping learning, only ARBs of the same class as ag are considered.

5 O-AIRS: an improvement of AIRS/AIRS2

In this section, we present O-AIRS, an innovative approach designed to refine Artificial Immune Systems (AIS) by addressing inherent limitations observed in traditional algorithms such as AIRS and AIRS2. Central to its improvements are two key concepts: Homogeneous Antigen Groups (HAGs) and a refined memory cell maturation mechanism.

AIRS algorithms faced two major hurdles: firstly, learning from individual antigens could lead to overfitting, where the system becomes overly influenced by random fluctuations in single data points. This skewed the development of memory cells, ultimately impacting classification accuracy. Secondly, not all memory cells were thoroughly evaluated during iterations. This results in a population of immature cells that couldn't handle diverse antigens, hindering classification performance.

5.1 Homogeneous antigen group

O-AIRS tackles the first issue by introducing HAGs. These represent groups of similar antigens belonging to the same class and residing close together within the data feature space. Unlike AIRS that dealt with single antigen one-at-a-time which may present noisy data [28], O-AIRS utilizes the collective information within HAGs during learning cycles.

This strategic shift reduces the impact of noise from isolated data points. By focusing on the average affinity between an antibody and an entire HAG, O-AIRS aims to make more robust and reliable classification decisions. Formula 4 presents the average affinity between an antibody and a HAG.

$$affinity(ac, GAH) = \frac{\sum_{i=1}^{n} affinity(ac,ag_i)}{n} \quad (4)$$

Where:

- ag_i is the i^{th} antigen in the group.
- n is the number of antigens in the group.
- ac is the antibody.

A parameter called GS (Group Scalar) controls the size and number of HAGs formed in O-AIRS. By adjusting GS, the system optimizes space allocation within the data space, ensuring cohesive HAGs with high internal similarity. This optimization aims to enhance classification accuracy by guaranteeing each HAG effectively represents a distinct antigenic subset.

Below, we propose a pseudo-code implementation (Algorithm1) for handling HAGs in O-AIRS.

```
initialize_param (affinity_threshold, GS);
all_antigen_group (0);
AGSet ← all_antigen ();
group ← 1;
foreach (antigen_ag ∈ AGSet) do
    if (antigen_ag.group = 0) then
        nearest_ag ← best_affinity (ag, AGSet);
        if (affinity (ag, nearest_ag) < affinity_threshold * GS) then
            ag.group ← group;
            group ← group + 1;
            go_to_next_ag ();
        end if
        if (nearest_ag.group=0 ) then
            ag.group ← group;
            nearest_ag.group ← group;
            group ← group + 1;
        else
            ag.group ← nearest_ag.group;
        endif
    endif
done
```

Algorithm 1: Homogeneous Antigen Group Handling.

5.1.1 Initialization

Before constructing HAGs, O-AIRS lays the foundation by initializing two key parameters:

- Affinity Threshold: this parameter defines the minimum affinity required between antigens for them to be grouped together into a HAG.
- Group Scalar (GS): GS acts as a pivotal factor influencing the formation of HAGs. A higher GS value typically results in fewer, larger HAGs, whereas a lower GS value tends to produce more, smaller HAGs.

Additionally, the function all_antigen_group (0) is employed to initially assign all antigens in the dataset to an ungrouped state, represented by the value 0.

5.1.2 Identifying neighbors

Next, O-AIRS examines each antigen and identifies its closest "neighbor" within the entire dataset using a function called best_affinity. This neighbor is the antigen with the highest affinity, essentially the most similar one.

5.1.3 Building communities

The algorithm then evaluates the affinity between the antigen and its nearest neighbor using Euclidean distance. This measure constitutes a critical parameter for artificial immune system algorithms [29].

Here is where the affinity_threshold and GS come into play:

High affinity (lower distance): if the affinity (Euclidean Distance) between the antigen and its neighbor is lower than the threshold defined by affinity_threshold * GS, they are considered "close enough" and grouped together into the same HAG. This indicates a high degree of similarity.

Low affinity (higher distance): if the affinity value exceeds the threshold, it suggests a lack of sufficient similarity. In this case, the current antigen becomes the founding member of a new HAG. This indicates a lower degree of similarity.

This process continues for all antigens in the dataset.

5.2 Ensuring memory cell maturity

O-AIRS goes beyond HAGs for classification. It implements a memory cell maturity mechanism to address the issue of incomplete testing in AIRS/AIRS2. The process involves two stages.

5.2.1 Initial selection with K-Nearest Neighbors (KNN)

O-AIRS utilizes the KNN algorithm to refine the data into a memory cell set (MC). This identifies memory cells with the closest affinity to incoming antigens, paving the way for accurate classification.

5.2.2 Interaction and maturation testing with HAGs

O-AIRS tracks memory cell interaction with HAGs. After antigen interaction, memory cells that haven't engaged with any HAG are categorized:

- Mature cells: these cells demonstrate sufficient HAG interaction and remain in the MC for future classification tasks.
- Learning set: memory cells lacking adequate HAG interaction are placed in a dedicated set for further training with updated antigenic data. This iterative learning process refines the classification capabilities of these immature cells.

The process of maturity test is presented in the flowchart in Figure 1.

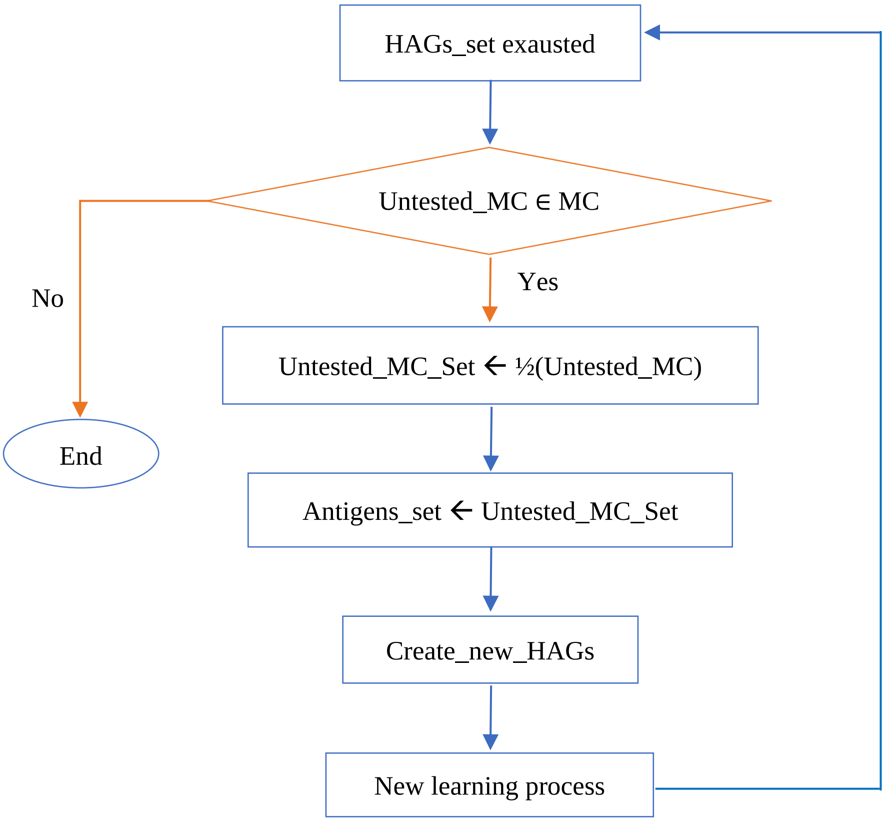


Figure 1: O-AIRS memory cell maturity Flowchart.

By leveraging HAGs and a refined memory cell maturity mechanism, O-AIRS aims to:

- Reduce memory footprint: by selectively retaining mature memory cells, O-AIRS optimizes memory usage, enhancing overall system efficiency.
- Enhance classification accuracy: through iterative interaction with HAGs and comprehensive memory cell testing, O-AIRS aims to achieve superior classification accuracy, even in noisy or complex datasets.
- Ensure comprehensive cell maturity: O-AIRS ensures all memory cells undergo rigorous testing and maturation, enhancing the system's resilience and adaptability in complex data environments.

6 Experimental study

This section presents an experimental comparison to evaluate O-AIRS, the improved version of AIRS and AIRS2. We implemented all three algorithms and tested their performance on two commonly used classification datasets: Haberman Surgery Survival (HSS) and Liver Disorders (LD).

6.1 Haberman surgery survival (HSS)

HSS is one of the hardest datasets for classification [30]. It contains case studies conducted between 1958 and 1970 at the University of Chicago Billings Hospital investigating the survival of patients who had undergone surgery for breast cancer. Characteristics of HSS dataset are depicted in Table 2.

Table 2: Characteristics of HSS dataset.

Number of attributes	Number of records	Number of positive records	Number of negative records
3	306	225	81

HSS dataset attributes are:

- Age of the patient during surgery
- Year of the surgery
- Number of axillary lymph nodes detected

HSS dataset class labels are:

- 1: Patient survived for 5 years or more
- 2: Patient died before 5 years

6.2 Liver disorder's (LD)

The Liver Disorders (LD) dataset contains 345 records of patients diagnosed with either confirmed liver disorders or no disorders. The characteristics of the LD dataset are presented in Table 3.

Table 3: Characteristics of LD dataset.

Number of attributes	Number of records	Number of positive records	Number of negative records
6	345	145	200

LD dataset attributes are:

- Mean Corpuscular Volume (MCV)
- Alkaline Phosphatase (ALP)
- Alanine Aminotransferase (ALT)
- Aspartate Aminotransferase (AST)
- Gamma-Glutamyl Transferase (GGT)
- Number of Alcoholic Drinks per Day

LD dataset class labels are:

- 1: Liver disorders
- 2: No liver disorders

6.3 Learning parameters

The initial learning phase involves creating memory cells (MC) and antigen recognition bodies (ARBs). The number of cells in these initial sets is determined randomly.

Subsequent learning steps utilize specific parameters defined in Table 4. These parameters influence the algorithm's behavior during the learning process.

Table 4: Learning parameters.

Parameter	Meaning	Domain of values	Chosen value
Hyper_clonal_rate	Maximum cloning rate	N	30
Clonal_rate	Average cloning rate	N	20
Mutate_rate	Mutation probability	[0,1]	0.1
Group_Scalar (GS)	Scale of distance between antigens in a HAG (O-AIRS parameter)	[0,1]	0.9

6.4 Results and discussion

Next, we evaluate O-AIRS against established algorithms on benchmark datasets (Liver Disorders and Haberman Surgery Survival) to assess its effectiveness in classification.

6.4.1 Experimental results with LD dataset

To compare the performance of AIRS, AIRS2, and O-AIRS, we used the LD dataset as input with unified common parameters. We assess the performance of the three algorithms using various metrics such as Accuracy, Precision, Recall, and F1-Score. These evaluation metrics are derived from confusion matrices, as shown in Tables 5, 6, and 7. The performance comparison of algorithms on the LD dataset is summarized in Table 8.

Table 5: Confusion matrix – AIRS/LD.

		Predicted	
		Liver disorders	No liver disorders
True	Liver disorders	47 (TP)	6 (FN)

	No liver disorders	27 (FP)	7 (TN)

Table 6: Confusion matrix – AIRS2/LD.

		Predicted	
		Liver disorders	**No liver disorders**
True	**Liver disorders**	50 (TP)	3 (FN)
	No liver disorders	25 (FP)	9 (TN)

Table 7: Confusion matrix – O-AIRS/LD.

		Predicted	
		Liver disorders	**No liver disorders**
True	**Liver disorders**	50 (TP)	3 (FN)
	No liver disorders	20 (FP)	14 (TN)

Table 8: Performance comparison of AIRS, AIRS2, and O-AIRS on the LD Dataset.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Number of memory cells used for classification**	**Percentage of tested memory cells**
AIRS	62,07%	63.51%	88.67%	74.01%	100	68%
AIRS2	67,82%	66.66%	94.33%	78.12%	100	70%
O-AIRS	73,56%	71.42%	94.33%	81.30%	50	100%

The experimental evaluation on the Liver Disorders (LD) dataset revealed compelling insights into the performance of AIRS, AIRS2, and O-AIRS in classification tasks. 8 summarizes the key metrics obtained from the experiments, focusing on Accuracy, Precision, Recall, F1-Score, the number of memory cells used for classification, and the percentage of tested memory cells.

Accuracy and classification performance: O-AIRS demonstrated a significant improvement in Accuracy compared to AIRS and AIRS2, achieving 73.56% Accuracy. In contrast, AIRS and AIRS2 achieved 62.07% and 67.82% Accuracy, respectively. This enhancement underscores O-AIRS's capability to more effectively classify instances within the LD dataset, highlighting its refined learning and activation mechanisms.

Precision, Recall, and F1-Score: Precision measures the proportion of correctly identified positive instances among all instances predicted as positive. O-AIRS exhibited a Precision of 71.42%, while AIRS and AIRS2 showed 63.51% and 66.66%, respectively. This indicates O-AIRS's ability to minimize False Positives more effectively. Similarly, O-AIRS achieved a Recall of 94.33%, outperforming AIRS (88.67%) and comparable to AIRS2 (94.33%). The F1-Score, which balances Precision and Recall, was highest for O-AIRS at 81.30%, compared to 74.01% for AIRS and 78.12% for AIRS2. These results demonstrate that O-AIRS not only enhances Precision but also maintains a high Recall rate, contributing to its superior F1-Score.

Memory cells utilization: a critical observation lies in the number of memory cells utilized for classification. O-AIRS effectively reduced the number of memory cells to 50, half that used by AIRS and AIRS2 (100 memory cells each). This reduction optimizes computational resources and enhances the efficiency of the classification process in O-AIRS, reflecting its streamlined and selective memory cell activation mechanism.

Tested memory cells: importantly, O-AIRS ensures that all memory cells retained for classification have been rigorously tested by antigens, achieving 100% activation fidelity. In contrast, AIRS and AIRS2 activated a lower percentage of memory cells (68% and 70%, respectively), potentially leading to less reliable classifications due to the presence of untested cells.

6.4.2 Experimental results with HSS dataset

To validate the findings from the LD dataset, the same evaluation process was conducted on the Haberman Surgery Survival (HSS) dataset. The algorithms (AIRS, AIRS2, and O-AIRS) were again evaluated using performance metrics like Accuracy, Precision, Recall, and F1-Score. Similar to the LD dataset analysis, these metrics are derived from confusion matrices (presented in Tables 9, 10, and 11). Table 12 presents a comprehensive comparison of the algorithms' performance metrics on the HSS dataset

Table 9: Confusion matrix – AIRS/HSS.

		Predicted	

		Patient died before 5 years.	**Patient survived for 5 years or more**
True	**Patient died before 5 years.**	60 (TP)	3 (FN)
	Patient survived for 5 years or more	10 (FP)	4 (TN)

Table 10: Confusion matrix – AIRS2/HSS.

		Predicted	
		Patient died before 5 years.	**Patient survived for 5 years or more**
True	**Patient died before 5 years.**	62 (TP)	1 (FN)
	Patient survived for 5 years or more	10 (FP)	4 (TN)

Table 11: Confusion matrix – O-AIRS/HSS.

		Predicted	
		Patient died before 5 years.	**Patient survived for 5 years or more**
True	**Patient died before 5 years.**	63 (TP)	1 (FN)
	Patient survived for 5 years or more	9 (FP)	4 (TN)

Table 12: Performance comparison of AIRS, AIRS2, and O-AIRS on the HSS Dataset.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Number of memory cells used for classification**	**Percentage of tested memory cells**
AIRS	77,92%	85.71%	95.23%	90.22%	100	58%
AIRS2	85,71%	86.11%	98.41%	91.85%	115	54,78%
O-AIRS	87,01%	87.50%	98.43%	92.64%	41	100%

Turning to the HSS dataset, Table 12 outlines the comparative performance of AIRS, AIRS2, and O-AIRS in terms of accuracy, precision, recall, F1-Score, the number of memory cells used for classification, and the percentage of tested memory cells.

Accuracy and classification performance: O-AIRS demonstrated robust classification Accuracy at 87.01%, surpassing both AIRS (77.92%) and AIRS2 (85.71%). This improvement underscores O-AIRS's efficacy in accurately predicting patient survival based on surgical outcomes in the HSS dataset.

Precision, Recall, and F1-Score: analyzing Precision, O-AIRS achieved 87.50%, outperforming AIRS (85.71%) and AIRS2 (86.11%). This higher Precision indicates O-AIRS's superior ability to identify true positive cases among all predicted positives. Moreover, O-AIRS maintained a commendable Recall of 98.43%, higher than AIRS (95.23%) and comparable to AIRS2 (98.41%). The F1-Score for O-AIRS stood at 92.64%, highlighting its balanced performance in Precision and Recall, essential for accurate classification in medical datasets.

Memory cells utilization: similar to its performance on the LD dataset, O-AIRS significantly reduced the number of memory cells utilized for classification in the HSS dataset to 41, compared to 100 for AIRS and 115 for AIRS2. This reduction not only optimizes computational efficiency but also enhances the interpretability and reliability of classification results in O-AIRS.

Tested memory cells: all memory cells retained for classification in O-AIRS were rigorously tested by antigens, ensuring 100% activation fidelity. In contrast, a substantial percentage of memory cells remained untested in AIRS (58%) and AIRS2 (54.78%), potentially leading to less reliable classifications and inconsistent performance.

6.4.3 Discussion and comparison

The experimental evaluation on both LD and HSS datasets establishes O-AIRS's superiority over AIRS and AIRS2. This is achieved through two key innovations: Homogeneous Antigen Groups (HAGs) and optimized memory cell utilization.

HAGs enhance pattern recognition by organizing antigens into groups based on similarity. This strategic grouping minimizes redundancy and improves classification accuracy by concentrating memory cell activations on pertinent antigenic features. Consequently, O-AIRS achieves superior performance while simultaneously reducing computational complexity.

Moreover, O-AIRS introduces a rigorous testing protocol for memory cells, ensuring that each retained cell undergoes thorough validation by antigens. This meticulous testing process contrasts sharply with traditional methods where many memory cells remain untested, potentially compromising the reliability of classifications. By attaining 100% activation fidelity through tested memory cells, O-AIRS enhances the robustness of its classifications and elevates overall detection rates in diverse datasets.

O-AIRS transcends its role as a mere enhancement over AIRS and AIRS2, positioning itself as a formidable and versatile competitor in the expansive field of classification. By leveraging its innovative mechanisms, O-AIRS effectively competes with other AINs. Traditional Artificial Immune Networks (AINs), such as AINet [11] and Adaptive Artificial Immune Networks (AAIN) [12], excel in tasks such as data clustering and classification but face challenges like computational intensity and sensitivity to parameter settings. Similarly, conventional Negative Selection Algorithms (NSAs) like Self-Nonself Discrimination [5], INSA [6], and ADC-NSA [7] demonstrate proficiency in specific tasks but encounter issues such as high computational costs and the need for precise parameter tuning. Algorithms grounded in Danger Theory, such as DeepDCA [16] and Multi-Level Intrusion Detection System [17], prioritize specific domains and achieve high accuracy rates but are often limited by dataset size and varying detection probabilities. Clonal Selection Algorithms (CSAs), represented by CLONALG [20] and CLONALG-M [23], excel in pattern recognition and optimization tasks but can be sensitive to parameter settings and computational overhead.

O-AIRS leverages Homogeneous Antigen Groups (HAGs) alongside a refined maturity mechanism to significantly boost data classification efficiency. By organizing antigens into coherent groups based on similarity, HAGs streamline the identification of complex data structures without the need for extensive parameter adjustments. This strategic approach not only enhances computational efficiency but also improves result accuracy by mitigating challenges such as uneven antigen distributions and the randomness inherent in detector generation. Consequently, O-AIRS emerges as a robust solution for applications in immune-inspired algorithms. Moreover, O-AIRS dynamically adapts to evolving data environments, thereby enhancing adaptability and robustness. The refined maturity mechanism, coupled with the effective implementation of HAGs, ensures optimal solutions are achieved efficiently. This capability is particularly advantageous in scenarios requiring rapid decision-making and resource efficiency, underscoring O-AIRS's versatility and practical utility across various domains.

7 Conclusion

In conclusion, while current Artificial Immune Systems (AIS) algorithms such as AIRS and AIRS2 offer a distinctive approach to machine learning, their limitations compromise their efficacy. These include reliance on single antigen activation and the retention of untested memory cells, resulting in inaccurate classifications.

This paper introduces the Optimized Artificial Immune Recognition System (O-AIRS), a novel solution aimed at addressing these shortcomings. O-AIRS integrates advanced functionalities such as Homogeneous Antigen Groups (HAGs) and a refined memory cell activation mechanism to enhance its classification capabilities. By leveraging HAGs, which delineate specific subsets within a class, O-AIRS optimizes memory cell activation tailored to precise threat recognition. Moreover, O-AIRS selectively engages validated memory cells during classification, mitigating concerns about untested recruits and facilitating more precise classifications.

O-AIRS's effectiveness was rigorously evaluated using established benchmark datasets. Comparative analyses against its predecessors, AIRS and AIRS2, across multiple metrics including accuracy, precision, and recall consistently demonstrated O-AIRS's superior classification performance. Additionally, O-AIRS exhibited robust adaptability to diverse datasets while maintaining computational efficiency, affirming its utility in complex real-world applications.

This efficiency is further underscored by O-AIRS's optimized activation mechanism, ensuring activation of only pertinent memory cells during classification and reducing the required number of cells by approximately 50% compared to original algorithms. Critically, all memory cells retained in O-AIRS are validated, enhancing their contribution to accurate classification outcomes.

Looking ahead, O-AIRS presents promising avenues for future research. One compelling direction involves exploring O-AIRS's dynamic adaptation of HAGs during classification, enhancing its ability to detect evolving threats in real-time. Such adaptive strategies hold particular promise in dynamic domains like cybersecurity and financial anomaly detection.

Furthermore, expanding O-AIRS's applicability beyond current datasets would provide insights into its generalizability and suitability for broader deployment. Exploring diverse distance metrics for antigen comparisons within HAGs could also yield additional performance enhancements. Additionally, investigating alternative memory cell types or hybrid approaches within O-AIRS offers potential for further augmenting its efficacy in handling intricate classification tasks.

References

[1] A.B. Watkins, AIRS: A resource-limited artificial immune classifier, Mississippi State University, 2001.

[2] A. Watkins, J. Timmis, Artificial immune recognition system (AIRS): Revisions and refinements, AISB 2004 Convention, 2004, pp. 18.

[3] R. Abdelkhalek, Z. Elouedi, Optimized Evidential AIRS with Feature Selection and Genetic Algorithm, International Conference on Intelligent Systems Design and Applications, Springer, 2022, pp. 1088-1097. https://doi.org/10.1007/978-3-031-27440-4

[4] R.V. Melo, D.D. de Macedo, A.M. Dantas, C.L. de Bona, A novel immune detection approach enhanced by attack graph based correlation, 2019 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2019, pp. 1-6. https://doi.org/10.1109/ISCC47284.2019.8969772

[5] S. Forrest, A.S. Perelson, L. Allen, R. Cherukuri, Self-nonself discrimination in a computer, Proceedings of 1994 IEEE computer society symposium on research in security and privacy, Ieee, 1994, pp. 202-212. https://doi.ieeecomputersociety.org/10.1109/RISP.1994.296580

[6] Y. Ren, X. Wang, C. Zhang, A novel fault diagnosis method based on improved negative selection algorithm, IEEE Transactions on Instrumentation and Measurement 70 (2020) 1-8. https://doi.org/10.1109/TIM.2020.3031166

[7] C. Yang, L. Jia, B.-Q. Chen, H.-Y. Wen, Negative selection algorithm based on antigen density clustering, IEEE Access 8 (2020) 44967-44975. https://doi.org/10.1109/ACCESS.2020.2976875

[8] X. Song, F. Gao, Z. Chen, W. Liu, A negative selection algorithm-based identification framework for distribution network faults with high resistance, IEEE Access 7 (2019) 109363-109374. https://doi.org/10.1109/ACCESS.2019.2933566

[9] N.K. Jerne, Towards a network theory of the immune system, Ann. Immunol. 125 (1974) 373-389.

[10] F.J. Varela, A. Coutinho, Second generation immune networks, Immunology today 12 (1991) 159-166. https://doi.org/10.1016/S0167-5699(05)80046-5

[11] L.N. De Casto, F.J. Von Zuben, An evolutionary immune network for data clustering, Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks, IEEE, 2000, pp. 84-89. https://doi.ieeecomputersociety.org/10.1109/SBRN.2000.889718

[12] J.M. Vidal, A.L.S. Orozco, L.J.G. Villalba, Adaptive artificial immune networks for mitigating DoS flooding attacks, Swarm and Evolutionary Computation 38 (2018) 94-108. https://doi.org/10.1016/j.swevo.2017.07.002

[13] L. Zhang, Y. Wang, X. Zhou, W. Liu, Improved artificial immune algorithm based on dynamic shrinkage strategy, 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), IEEE, 2022, pp. 867-872. https://doi.org/10.1109/ICCECE54139.2022.9712692

[14] S. Kanwal, F. Khan, S. Alamri, K. Dashtipur, M. Gogate, COVID-opt-aiNet: A clinical decision support system for COVID-19 detection, International Journal of Imaging Systems and Technology 32 (2022) 444-461. https://doi.org/10.1002/ima.22695

[15] P. Matzinger, Essay 1: the Danger model in its historical context, Scandinavian journal of immunology 54 (2001) 4-9. https://doi.org/10.1046/j.1365-3083.2001.00974.x

[16] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, A. Al-Barakati, Deepdca: novel network-based detection of iot attacks using artificial immune system, Applied Sciences 10 (2020) 1909. https://doi.org/10.3390/app10061909

[17] V.T. Alaparthy, S.D. Morgera, A multi-level intrusion detection system for wireless sensor networks based on immune theory, IEEE Access 6 (2018) 47364-47373. https://doi.org/10.1109/ACCESS.2018.2866962

[18] H.M. Salmon, C.M. De Farias, P. Loureiro, L. Pirmez, S. Rossetto, P.H. de A Rodrigues, R. Pirmez, F.C. Delicato, L.F.R. da Costa Carmo, Intrusion detection system for wireless sensor networks using danger theory immune-inspired techniques, International journal of wireless information networks 20 (2013) 39-66. https://doi.org/10.1007/s10776-012-0179-z

[19] X. Zhou, K. Liu, Z. Jin, S. Tian, Y. Fu, L. Qin, Artificial Immune Based Cryptography Optimization Algorithm, Informatica 38 (2014).

[20] L.N. De Castro, F.J. Von Zuben, Learning and optimization using the clonal selection principle, IEEE transactions on evolutionary computation 6 (2002) 239-251. https://doi.org/10.1109/TEVC.2002.1011539

[21] R. Huang, H. Tawfik, A. Nagar, Licence plate character recognition using artificial immune technique, International Conference on Computational Science, Springer, 2008, pp. 823-832. https://doi.org/10.1007/978-3-540-69389-5

[22] E.D. Ülker, Gene Selection in Microarray Data Using an Improved Approach of CLONALG, The International Conference on Artificial Intelligence and Applied Mathematics in Engineering, Springer, 2019, pp. 466-472. https://doi.org/10.1007/978-3-030-36178-5

[23] S.A. Sert, A. Yazici, Increasing energy efficiency of rule-based fuzzy clustering algorithms using CLONALG-M for wireless sensor networks, Applied Soft Computing 109 (2021) 107510. https://doi.org/10.1016/j.asoc.2021.107510

[24] Z. Zareizadeh, M.S. Helfroush, A. Rahideh, K. Kazemi, A robust gene clustering algorithm based on clonal selection in multiobjective optimization framework, Expert Systems with Applications 113 (2018) 301-314. https://doi.org/10.1016/j.eswa.2018.06.047

[25] S. Shekhar, D. Sharma, D. Agarwal, Y. Pathak, Artificial immune systems-based classification model for code-mixed social media data, IRBM 43 (2020) 10. https://doi.org/10.1016/j.irbm.2020.07.004

[26] W. Nebili, B. Farou, Z. Kouahla, H. Seridi, Revised Artificial Immune Recognition System, IEEE Access

9 (2021) 167477-167488. https://doi.org/10.1109/ACCESS.2021.3133731

[27] A.B. Watkins, Exploiting immunological metaphors in the development of serial, parallel and distributed learning algorithms, University of Kent (United Kingdom), 2005.

[28] A.A. Freitas, J. Timmis, Revisiting the foundations of artificial immune systems for data mining, IEEE Transactions on Evolutionary Computation 11 (2007) 521-540. https://doi.org/10.1109/TEVC.2006.884042

[29] H. Merabti, B. Farou, H. Seridi, A segmentation-recognition approach with a fuzzy-artificial immune system for unconstrained handwritten connected digits, Informatica 42 (2018) 95-106.

[30] H.N.A. Pham, E. Triantaphyllou, An application of a new meta-heuristic for optimizing the classification accuracy when analyzing some medical datasets, Expert Systems with Applications 36 (2009) 9240-9249. https://doi.org/10.1016/j.eswa.2008.12.007

A Review on Deep Learning Techniques for EEG-Based Driver Drowsiness Detection Systems

Imene Latreche*1, Sihem Slatnia1, Okba Kazar2, Ezedin Barka3, Saad Harous4
1 Department of computer science, University of Mohamed Khider, Biskra, Algeria
2 College of Arts, Sciences & Information Technology, University of Kalba, Sharjah, UAE
3 Department of Information Systems and Security, UAE University, Al Ain, UAE
4 Computer Science Department, College of Computing and Informatics,
University of Sharjah, Sharjah
E-mail: Imene.latreche@univ-biskra.dz, sihem.slatnia@univ-biskra.dz, kazarokba@gmail.com, ebarka@uaeu.ac.ae, harous@sharjah.ac.ae
*Corresponding author

Keywords: driver drowsiness, driver fatigue, electroencephalogram (eeg), deep learning, detection, driver fatigue

Received: July 22, 2023

Driver Drowsiness is considered one of the significant causes of road accidents and fatal injuries. Due to this, creating systems that can monitor drivers and detect early drowsiness has become an important field of research and a challenging task in recent years. Several research attempts were proposed to solve this problem based on several approaches and techniques. The Electroencephalogram (EEG) is one of the most efficient and reliable method, among the physiological signals-based monitoring approaches. In this area, many Machine Learning (ML) techniques have been used to detect EEG-based driver drowsiness. However, due to the limitations of ML techniques, many researchers have shifted their focus to the use of deep learning (DL) techniques, which have demonstrated superior performance in many fields including the physiological signals classification tasks. This paper reviews and discusses numerous new research papers that have proposed and implemented driver drowsiness detection systems based on EEG and deep learning techniques. In addition, we have outlined the limitations and difficulties of the existing works and highlighted and proposed some propositions that will help future field researchers enhance and generalize the results. Based on our thorough analysis, we have determined that the latest advancements in detecting driver drowsiness have employed the convolutional neural network (CNN) technique, which has demonstrated effective performance in classifying signals. Furthermore, the primary issue encountered in all works is developing a more precise and accurate method. Nevertheless, we seek a precise system capable of swiftly identifying a state of drowsiness while using minimal spatial memory and processing resources.

Povzetek: Narejen je pregled objav za zaznavanje zaspanosti voznikov na osnovi EEG signalov z uporabo metod globokega učenja. CNN se izkaže za učinkovito metodo pri klasifikaciji signalov.

1 Introduction

A significant number of individuals face extended periods of work during the night, including security officers, truckers, and medical personnel. Consequently, operating a motor vehicle while experiencing fatigue is widespread. It is likely something that the majority of drivers have done. It is crucial to develop a method for alerting drivers when their fatigue reaches a critical level, hindering their ability to drive safely.

Drowsiness, known as fatigue, is a psychophysiological transition state between alertness and sleep. When a driver is in this state, his/her concentration and performance decrease, while his/her reaction time increases [1]. This state affects the results of some tasks requiring concentration, such as driving [2]. Driver drowsiness is the third cause of traffic accidents, and is responsible for 25% of road accidents, following high speed and alcoholism [3]. Studies have demonstrated that 24-hour sleep deprivation induces the same degree of impairment as an individual with a blood alcohol concentration of 0.10%, which exceeds the legal limit. Based the National Highway Traffic Safety Administration (NHTSA), drowsy driving has caused 100000 crashes, more than 1500 deaths, and $12.5 billion in monetary losses [4]. Nevertheless, the National Highway Traffic Safety Administration (NHTSA) acknowledges the difficulty in accurately quantifying the exact figures of accidents, injuries, or fatalities caused by drowsy driving. It recognizes that the reported statistics are lower than the actual occurrences. According to a study by the Foundation for Traffic Safety of the American Automobile Association, over 320,000 drowsy driving accidents occur annually, including 6400 crashes resulting in fatalities [56]. The statistics indicate that driver drowsiness is a real problem; therefore, it is necessary to develop a system that can detect it quickly and accurately in its early stages and alarm the driver to prevent road accidents and reduce the fatality rate. In this regard, several researchers have proposed numerous approaches to detect drowsy drivers. These approaches can be categorized into three main categories: vehicle-based, information obtained by monitoring the vehicle's movement and behavior; behavior-based, information obtained through the analysis of the driver's facial expressions and movements; and physiological signals-based obtained by attaching specialized sensors to the driver's body as summarized in Figure 1 [5]. Physiological methods are the most used because they have proven to be effective, accurate, and reliable [5, 6]. Physiological signals such as Electrocardiogram (ECG), Electrooculogram (EOG), Photoplethysmogram (PPG), Electromyogram (EMG), and Electroencephalogram (EEG) were utilized to identify drowsy drivers because, they can detect the body signals changes and compare this change to the normal state [1].

The Electroencephalogram (EEG), a record of the electrical activities of different brain regions [1], is the most widely used of the Physiological signals. It is known as the gold standard for drowsiness detection due to its low cost, usability, and dependability [6]. EEG signals are measured and captured by placing a device containing a pattern of electrodes on the scalp based on the international 10-20 system of EEG electrode placement. These signals are subdivided into several bands based on frequency. There are five well know bands: the delta band (0.5-3 Hz), the theta band (4-7 Hz), the alpha band (8- 13 Hz), the beta band (14-30 Hz), and gamma-band (greater than 30 Hz) [8].

To detect drowsiness state successfully, many researchers have proposed and implemented robust detection systems using two well-known mechanisms, Machine Learning (ML) and Deep Learning (DL). Machine Learning is a subfield of artificial intelligence. It has been used in several classification tasks [9]. Hu & Min. [10] have proposed a gradient boosting decision Tree (GBDT) to determine whether a driver is drowsy or not. They claim the accuracy reached 94%. In [11], Mu et al. used the SVM algorithm to classify a driver as tired or awake based on the forehead FP1 and FP2 electrodes. The accuracy of this approach is 85%. the following five ML techniques: the K-nearest neighbor (KNN), support vector machine (SVM), extreme learning machine (ELM), hierarchical extreme learning machine (H-ELM), and the modified hierarchical extreme learning machine algorithm with particle swarm optimization (PSO-H-ELM) have been proposed in [12] to identify the drowsiness state using EEG signals. The achieved accuracy of these algorithms is 79.31%, 79.31%, 74.08%, 81.67%, and 83.12%, respectively. Nevertheless, ML techniques have limitations, such the need for massive data and hand-crafted feature extraction as intermediary steps [3] (see Figure 2).

Deep learning is a subfield of machine learning that has been utilized in different fields, including speech recognition, computer vision, and natural language processing [13]. DL techniques have demonstrated their effectiveness in EEG task classification, especially convolutional neural network (CNN), because it does not require hand-crafted feature extraction. They can automatically detect and learn features through convolutional layers [14, 5].

This review aims to present and explain the pipeline of an EEG-Based driver drowsiness detection system using deep learning techniques. Then, analyze and discuss several new research papers that used deep learning techniques to detect and classify whether a driver is in a drowsy or awake state using EEG signals, especially those published over the past three years, by listing the extracted features, methods, classifiers, and the used datasets, and classification metrics such as accuracy, sensitivity, and precision. Finally, highlight the limitations and challenges of the reviewed papers and propose future improvements. The paper is organized as follows: The second section outlines the search strategy followed in this work. The third section describes the backgrounds and the related works cited in this literature review. The results are discussed in the fourth section. The fifth section presents the challenges and limitations of the discussed works furthermore some propositions that can be as future works. Finally, a conclusion is provided in the sixth section.

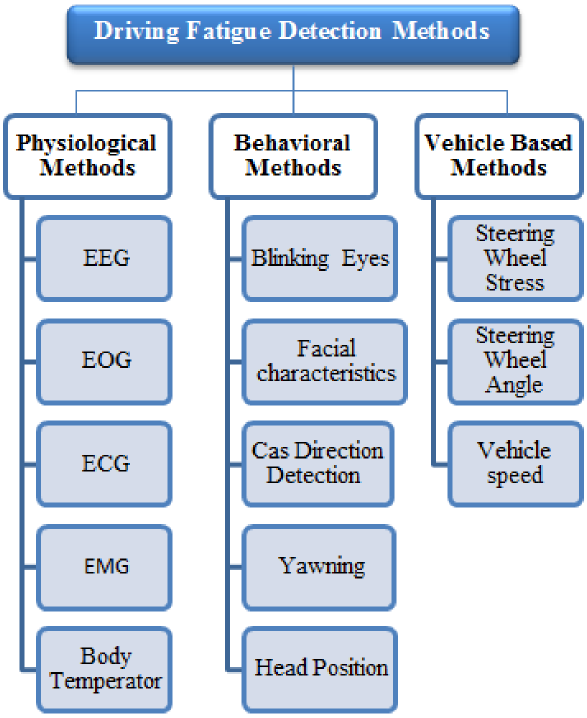


Figure 1: Drivers’ fatigue detection methods [7]

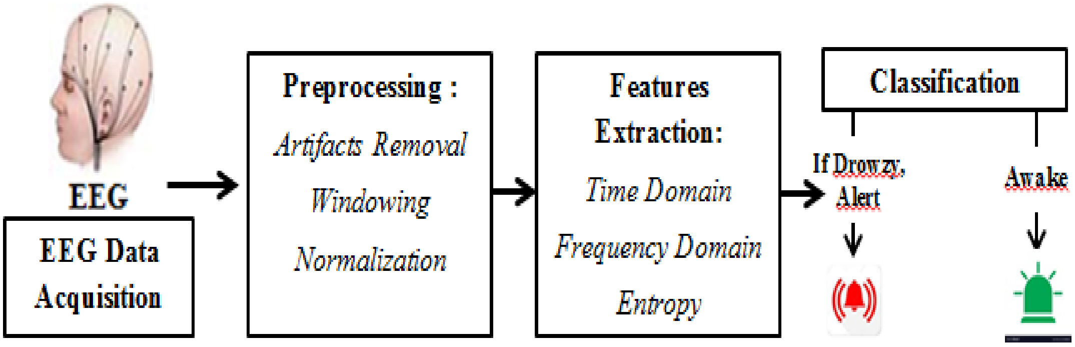


Figure 2: EEG based drowsiness detection and warning scheme [1]

2 Search strategy

The keywords used for collecting the papers are "Driver Drowsiness", "Driver fatigue" "EEG", "Electroencephalogram", "Deep Learning", "Detection", and the query used in Google scholar was "deep learning for EEG-based driver drowsiness detection system" and "deep learning for EEG-based driver fatigue detection system". Furthermore, the papers were selected based on three criteria the paper must be: written in English, either a journal article or conference article (reviews papers were excluded), and new (last three years).

Many researchers have employed deep learning techniques to detect driver drowsiness based on EEG signals. For example, Google Scholar results using this query "deep learning for EEG-based driver drowsiness detection system" have shown that from 2019 to 2022 (03/04/2023), about 5240 works were published in this context (see Figure. 3).

3 Deep learning for EEG-Based driver drowsiness

Figure 4 illustrates a general architecture of an EEG-based drowsiness detection system using deep learning techniques. The process starts by collecting EEG signals using one of the existing wearable devices placed on the scalp to acquire raw data; the obtained signals are then preprocessed to remove artifacts, normalize and prepare them for feeding into a DL model that classifies whether the individual is drowsy.

3.1 Data acquisition

The first step of an EEG-Based driver drowsiness system is acquiring and collecting Real-time EEG signals. However, because of safety concerns collecting real-time EEG is not feasible, leading many researchers to use driving simulators under several experimental protocols see Figure. 5.

After preparing the simulated environment, a wearable device, a set of electrodes placed on the scalp based on the international 10-20 system, must be used to acquire the EEG data. However, a wearable device with numerous electrodes is expensive and may be uncomfortable for the driver. Due to these limitations, some researchers have focused on identifying the most effective and informative regions that can provide more information about drivers' states using fewer electrodes [52].

3.2 EEG- based drowsiness datasets

A dataset is a collection of information on a specific subject that can be used by machine learning and deep learning methods for many purposes, such as classification and prediction. For example, there are various available online datasets on EEG-based drowsiness context.

3.2.1 Sleep -*EDF* dataset

The Sleep-EDF dataset [36] is obtained from the Physionet database [37], which contains 197 whole-night Polysomnographic (PSG) sleep recordings with EEG, EOG, chin EMG, and event markers. They were sampled at 100 Hz and 1 Hz.

3.2.2 The Original EEG data for driver fatigue detection

This dataset is generated by a 40-channel Neuroscan amplifier. It contains twelve healthy subjects and twelve drowsy subjects. The signals are obtained from a 32-channel electrode cap (30 effective channels and two reference channels), and digitized at 1000 Hz [38, 42].

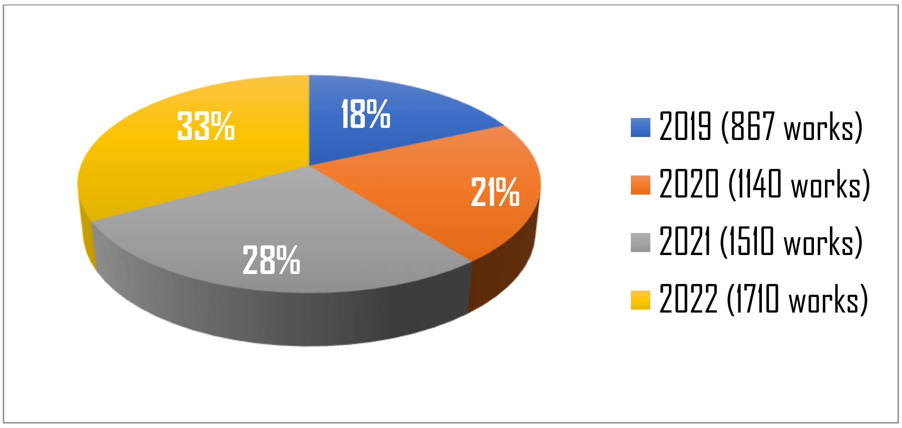


Figure 3: Number of studies using deep learning for EEG-driver drowsiness detection during 2018 -2023

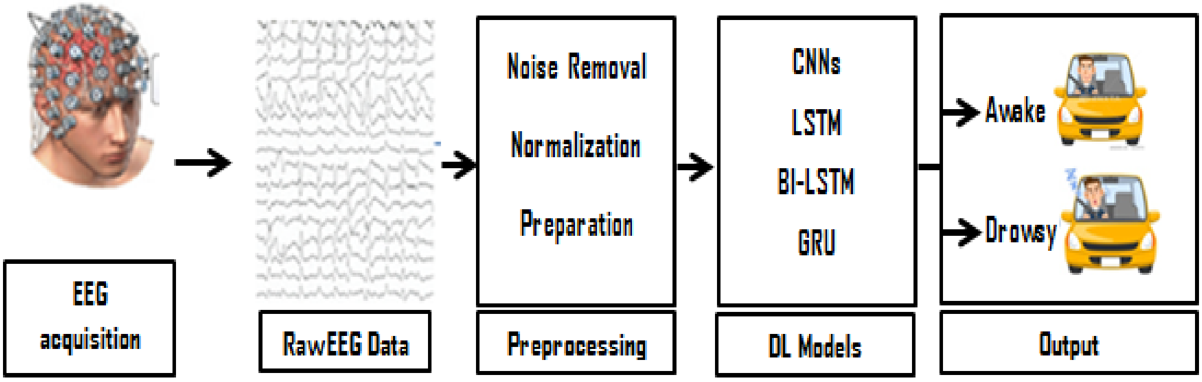


Figure 4: General architecture of a DL based EEG drowsiness detection system

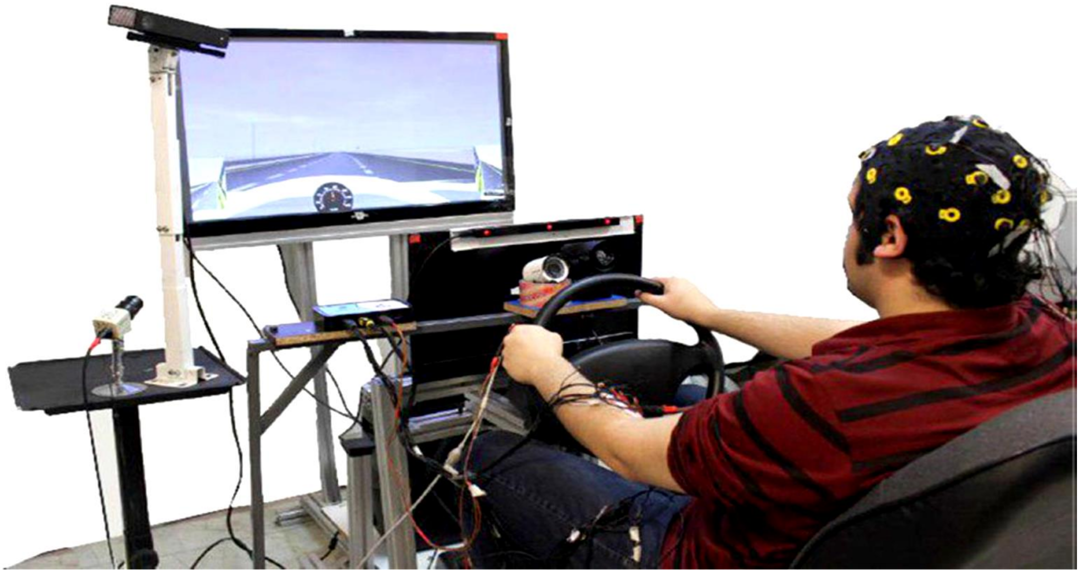


Figure 5: The road scene of the driving simulator [5]

3.2.3 Multi-channel EEG recordings during a sustained-attention driving task

This dataset consists of twenty-seven subjects that participate in a 90-minute driving task from the National Chiao Tung University. The signals were acquired using a 32 Ag/AgCl electrodes EEG wired cap (two reference electrodes) and digitized at 500 Hz. The Institutional Review Broad of Taipei Veterans General Hospital, Taiwan, had approved the experimental protocol [39].

3.2.4 MIT/*BIH Polysomnographic EEG database*

The database is collected from 16 male subjects using the C3-O1, C4-A1, and O2-A1 EEG channels. The database contains 18 records, each with four files. The physiological signals were digitized at a sampling rate of 250 Hz [37, 40].

3.2.5 SEED-VIG dataset

The SEED-VIG dataset is designed to investigate the vigilance estimation problem. It was collected from 23 participants and lasted approximately 2 hours. The Dataset is acquired using 18 electrode channels according to the international standard 10-20 system and down-sampled to 200 Hz. The records were labeled using the SMI eye-tracking glasses with the PERCLOS indicator [41].

Table 1: Publicly available EEG dataset for driver fatigue

Dataset	Subjects	Electrodes	Sampling frequency
Sleep-EDF [36, 37]	/	Fpz-Cz / Pz-Oz.	100 Hz
The original EEG data [38, 42]	12	32	1000 Hz
Multi-channel EEG recordings [39]	27	32	500 Hz
MIT/BIH Polysomnographic EEG [37, 40]	16	C3-O1, C4-A1, and O2-A1channels	250 Hz
SEED-VIG [41]	23	18	200 Hz

3.3 Preprocessing

EEG signal preprocessing is an essential step that can be defined as a set of signal processing steps that transform raw EEG data into a more suitable form that can be easily analyzed and handled [22]. The preprocessing involves three steps (see Figure 6): 1) removing the noise and artifacts to get closer to real neural signals, 2) normalization to scale the values of all EEG signals, and 3) signal preparation or EEG analysis.

For noise removal, many methods can be used. The most popular ones are finite impulse response (FIR) and infinite impulse response (IIR) filters. There exist various techniques, such as the min-max scaling, robust scaling, standard scaling and the z-score technique that can be used during the normalization step. Finally, different time domain, frequency, and time-frequency methods are employed to prepare the signals to be fed to a DL model, such as Fourier transform techniques, Component Analysis, and Wavelet Transform [23].

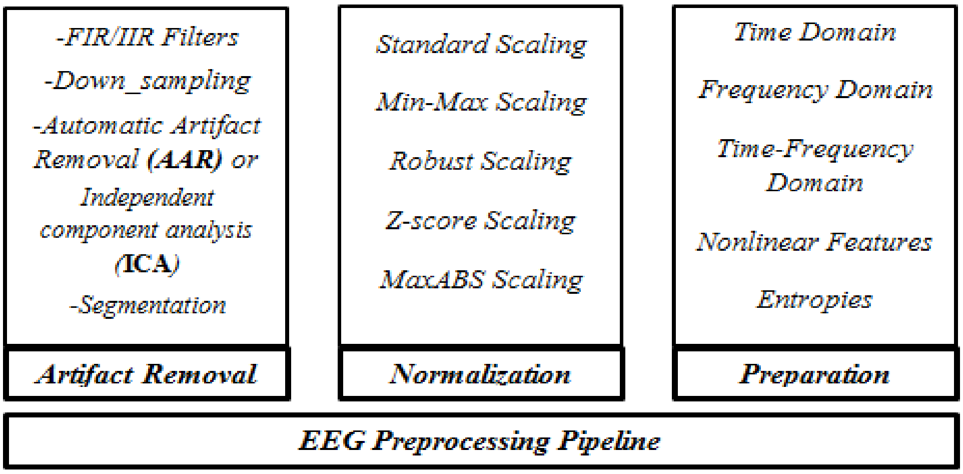


Figure 6: The EEG preprocessing pipeline

3.4 Review of deep learning techniques

Deep learning (DL) is one of the main techniques of machine learning. Deep Learning techniques are algorithms capable of emulating the human brain's actions using artificial neural networks. These artificial neural networks are constructed of tens or hundreds of neuron layers. Each layer receives and interprets information from the previous layer [24]. No human intervention is required for DL; however, a large amount of data is needed to map the given input to specific labels [25]. DL allows feeding deep neural networks DNNs with the raw data with limited or no preprocessing. In addition, in DL, the feature extraction, selection, and classification are constructed as a single pipeline [26].

DL techniques can be categorized into three main groups: deep networks for supervised learning, deep networks for unsupervised learning, and deep networks for hybrid learning. In the first category, "supervised learning", there are three main techniques Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). In the second category, "unsupervised learning" we find the Generative Adversarial Network (GAN) and Auto Encoder (AE) and Its Variants [27].

3.4.1 Convolutional neural networks (CNN)

The Convolutional Neural Network (CNN) is one of the most popular supervised deep learning architectures that learn directly from the input without the need for handcraft feature extraction [27]. The basic CNN is like the multi-layer perceptron (MLP). It consists of many convolution layers that are followed by sub-sampling (pooling) layers that precede the last fully connected FC layers [25]. Initially, CNN was designed for image classification (two-dimensional input); but nowadays, it is also used to classify one-dimensional (1D) data such as biological signals (ECG, EMG, EEG...).

A. 2D Convolutional Neural Networks (2D-CNNs)

CNN is a deep learning model that is considered the most used, especially to deal with 2D shapes like images; therefore it is often called 2D-CNN. It is used in other fields such as natural language processing and visual recognition [27]. CNN has many different variants, such as VGG, AlexNet, GoogleNet, and ResNet. Each one of these variants has a specific architecture. They have been employed in many fields [27]. Apostolopoulos and Tzani [30] have proposed VGG19 and the MobileNet v2 to detect Covid 19 from X-ray images. In [31], M. Hussain et al. have used the Inception-v3 on both the Caltech Face dataset and the CIFAR-10 dataset.

The 2D-CNN has also been used to classify signals such as EEG signals for the diagnosis of neuronal disorders such as epilepsy and driver fatigue by transforming them into a two-dimensional spectrum (2D spectrogram) using various methods (continuous wavelet transform (CWT) [28] and Short-time Fourier transform (ST) [7]. Figure 7 presents the architecture of a 2D-CNN used to classify EEG signals into two classes “Normal” or “Seizure”.

B. 1D Convolutional Neural Networks (1D-CNNs)

Recently, a 1D-CNN is proposed to deal with 1D signal and data repositories. It demonstrated excellent performance in many fields, including biomedical data classification [29] and EEG classification [5]. 1D-CNN is a modified version of 2D CNNs that use 1D convolution operation (scalar multiplications and additions). One of its significant advantages is that in terms of computational complexity, it is lower than the 2D-CNN [29].

Figure 8 presents the architecture of 1D-CNN that can be used for EEG signals classification and seizures detection.

3.4.2 Recurrent neural networks (RNN)

Another type of DL technique, RNN, is employed mostly to deal with time-series or sequential data such as signals, text, and videos. It is used often in natural language processing and speech recognition. RNN feeds the output of the previous step as an input to the current step; that is known as a circulation behavior. It requires integral memory cells that preserve the previous outputs, whereas the integrated memory cell has three gates titled input, output, and forget gates. The main challenge of the standard recurrent neural networks is learning long data sequences because of the vanishing issues gradients. The Long short-term memory (LSTM) and Gated recurrent units (GRUs) are RNN models used to minimize those issues, and that perform well in many domains and real-world applications [26, 27].

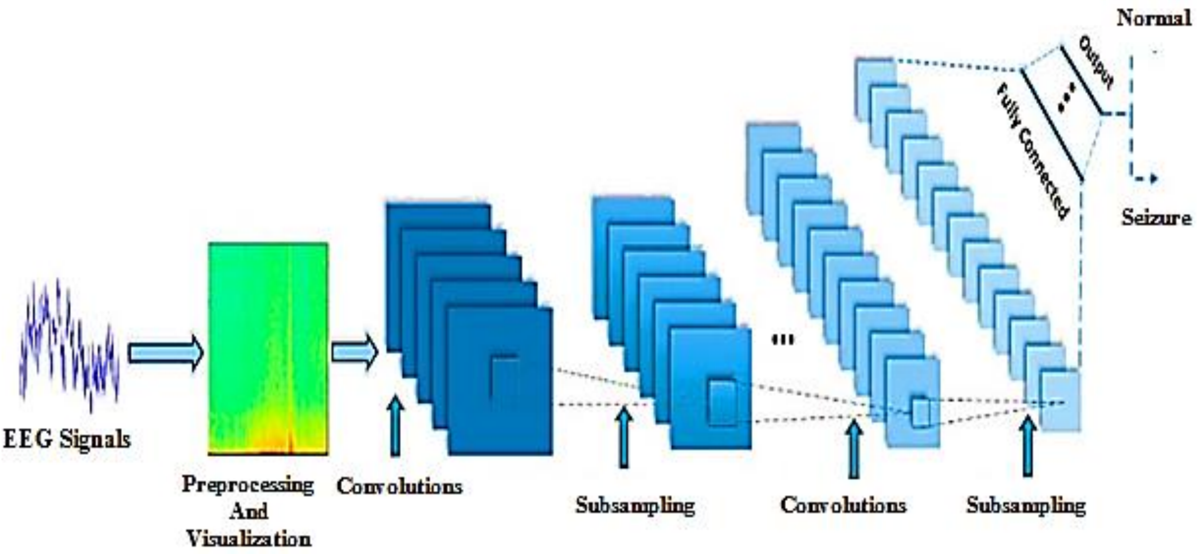


Figure 7: A typical 2D-CNN for EEG classification [23]

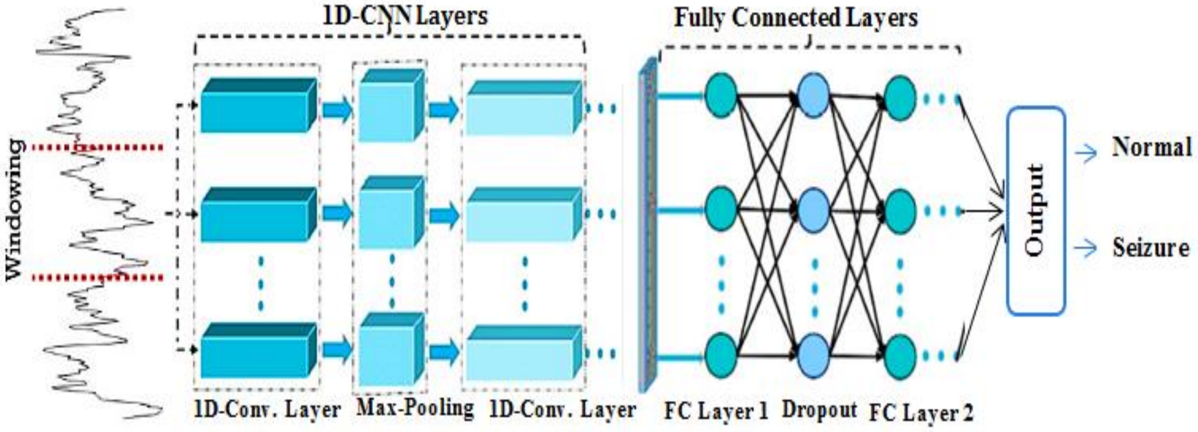


Figure 8: A typical 1D-CNN that can be used for EEG classification [23]

A. Long short-term memory (LSTM)

The Long Short-Term Memory (LSTM) is an enhanced version of the Recurrent Neural Network (RNN) that addresses the gradient vanishing problem and long-term temporal dependencies. The LSTM layer is distinguished by memory blocks, which are hidden units [32]. Each memory block consists of recurrently connected memory cells, with each cell containing weights and three gates, "the input, forget, and output gates," which are the distinguishing feature of LSTM models (see Figure. 9) [33, 34].

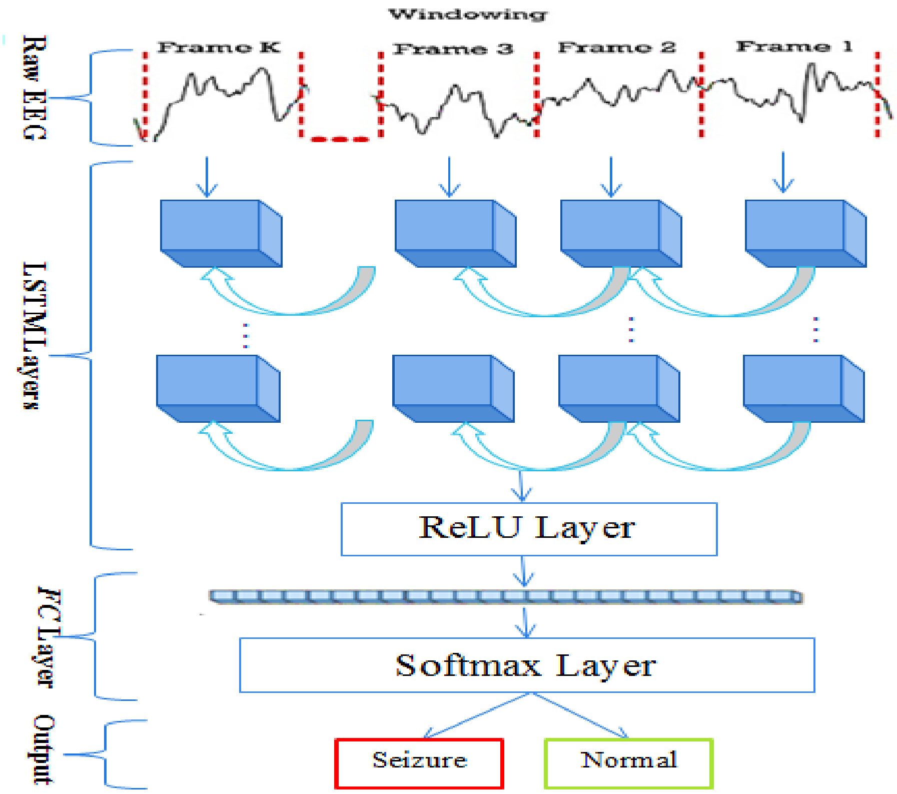


Figure 9: A typical LSTM that can be used for EEG classification

B. *Gated recurrent units (GRUs)*

A Gated Recurrent Unit (GRU) is another popular variant of the recurrent network. It is considered a simpler version of LSTM. GRU includes gating units that modulate the flow of information inside the unit without any separate memory cells. However, instead of three gates, the GRU has only two: the reset gate and the update gate [35].

3.5 EEG-Based driver drowsiness systems

In this Section we review and analyze some recent EEG-Based drowsiness detection schemes. Tables 2 and 4 summarized the reviewed works.

Houshmand et al. [5] designed three CNNs to detect driver drowsiness in the early stage based on single-channel signal and EEG alpha spindles. The first one is a 1D CNN with three convolution layers, three max-pooling layers, and two fully connected layers. The second one is a 2D CNN with four convolution layers, four max-pooling layers, one fully connected layer, and one flattens layer. The third one is identical to the second but with different parameter values (number of filters and nodes). The activation function of the three models is dropout. The fed data to the models are raw EEG data for 1D CNN, power spectrum analysis EEG data for the second CNN, and the CWT of EEG epochs for the third model. The P4 channel is determined to have the highest feature weight for drowsiness classification based on the neighborhood components analysis technique. The best results are obtained by CWT-CNN as follows: accuracy 94%, recall 95%, precision 91%, and F1-score 93%.

The author in [7] proposed a driver fatigue detection system based on a single channel EEG and transfer learning. The system works as follows: the acquired EEG raw data is passed through a preprocessing pipeline, and then the processed data is transformed into a two-dimensional spectrum using the Short-time Fourier Transform (STFT). After that, the resulting spectrum is then fed to a modified AlexNet CNN model employing transfer learning to determine whether the driver is drowsy or alert. The author applied transfer learning using fine-tuning to the original AlexNet by replacing the final classification layer with another layer and reducing the last fully connected layer nodes number to 5. To find the best channel, the author compared the accuracy obtained by the AlexNet CNN of seven different channels (FP1, FP2, T3, T4, O1, O2, and Oz). The FP1 and T3 channels achieved the best accuracy of 90% and 91%, respectively.

Cheng et al. [18] have developed an EEG-based prediction system to estimate the drowsiness level of drivers. They use a raw EEG signal without utilizing any artifact removal methods. The 256-point fast Fourier transform (FFT) is used to transform the time-series EEG data into the frequency domain (an image-like feature map). In the final stage, the EEG images are passed into a CNN to classify them into two classes “Drowsy” or “Awake”. The used CNN is a basic CNN with two convolutional layers, two max-pooling layers, and one fully connected layer. Three training data sets are evaluated using a leave-one-subject-out cross-validation strategy during the training phase. The proposed work outperforms the SVM classifier in both balanced (69.18 %) and imbalanced (71.15 %) data.

Guarda et al. [44] used convolutional neural networks (CNNs) to classify whether a driver is in a drowsy state or not based on EEG spectrograms. The proposed method is applied to the ULg Multimodality Drowsiness Database, where only the Fz and Pz channels are used to train the model. First, the raw EEG signals is transformed into spectrograms that have been converted into a gray-scale format. After the preprocessing and the generation of the gray-scale spectrograms, six sets of spectrograms are obtained. The proposed CNN has three convolutional layers, three pooling layers, and one fully connected layer. The input data are one or two 96*96 images (depending on how many EEG sensors are used). The proposed model achieved an accuracy of 86.74% with Fz-Pz channels and 13 seconds signals which is the best compared to the other sets. The model outperformed SVM, NN, and RF in all metrics.

Gao et al. [48] proposed a recurrence network-based convolutional neural network (RN-CNN) method to detect driver drowsiness using EEG signals. First, they collected EEG signals from 10 subjects and 30 channels using a simulated driving experiment and pre-processed the obtained signals. Then they used a recurrence network (RN), a complex network method that transforms raw EEG into a mutual information matrix of 30*30. Finally, they fed the mutual information matrix into CNN architecture to extract features and classify the driver state. The used CNN has two convolutional layers, two fully connected layers, and a softmax layer. The proposed approach achieved an accuracy of 92.25%. To evaluate the performance of the RN-CNN, the authors compared its result with other state-of-art works such as FT-CNN, PSD-SVM, CSP-SVM, and others. RN-CNN outperformed all the considered protocols.

Chaabene et al. [3] proposed an EEG-based CNN Driver Drowsiness system. Data acquisition and model analysis are the two primary procedures of the proposed architecture. The data acquisition step is divided into two-part: data collection using the Emotiv EPOC+ headset to record 14 channels and preprocessing. Data preparation to remove noise and artifacts, data annotation, and data augmentation to prevent overfitting and improve accuracy are parts of the preprocessing step. In the second step, "model analysis" a CNN with four convolution layers, one max-pooling layer, and two fully connected layers, is implemented. Two experiments were conducted to evaluate the system's performance using two classes (Drowsy/Awake); the first was conducted without data augmentation, with 2, 4, 7, and 14 channels, and the best accuracy (79.43%) is achieved with 14 channels. However, data augmentation is used in the second experiment to reach 90.14% accuracy using seven channels.

Balam et al. [6] have proposed CNN architecture for automated Driver Drowsiness detection based on a Single-Channel EEG signal. The used data are raw EEG signals from the physionet Pz-Oz dataset. To determine the best CNN model. The authors evaluated the performance of numerous CNN models with different kernel sizes, varying numbers of hidden layers, and multiple sets of filters. The proposed model is created by combining the best two evaluated CNN (CNN [4HL, 9F, 3KS], CNN [3HL, 3F, 5KS]). For the evaluation of the model's performance, three different training strategies were utilized: subject-wise, cross-subject-wise, and combined-subjects-wise validations. The results indicate that the highest accuracy 94%) was achieved with combined subjects. A comparative study is conducted between the proposed model and other models that utilized the same dataset, and the results demonstrated that the employed CNN achieved the highest accuracy of 94.87%. However, the results were close.

Ding et al. [15] have implemented a Deep Learning architecture on a mobile device that uses a single-channel EEG signal to Detect Driver Drowsiness. Their study aims to get high accuracy with a small model size and predict latency compared to the existing models. The components of the proposed architecture are an EEG signal collector, a trained model integrated into a Smartphone to predict the state of the driver then alert him/her, a cloud database that serves as a backend, and a web page that contains a remote monitor to observe the real-time condition and historical record from the backend. The employed model is a Cascaded CNN with an attention mechanism layer; it is made of three blocks: Dimension Reduced Level, Feature Extract Level, and Full Connect Level. In terms of accuracy and recall, the proposed model was compared to other deep learning and machine learning architectures. The latter outperformed the others with an accuracy of 97.26% and a recall of 96.56%. In addition, the model size (1.61 MB) and latency (26s/epoch) are more suitable for a real-time mobile system.

The authors in [16] developed a system capable of detecting vehicle driver drowsiness using a wearable EEG device and CNN. The proposed system consists of a wearable device to acquire EEG signals, a preprocessing step to remove artifacts and improve the information's quality, a trained model based on CNN for signal classification, and, as a final step, an early warning strategy to restore the driver attention. Two models were used for the classification step: a CNN with an Inception module containing five convolutional layers, two pooling layers, three Inception modules, and three fully connected layers, and the Modified AlexNet model that includes eight convolutional layers, four pooling layers, and three fully-connected layers. The obtained accuracy of the inception model is 95.69%, which is greater than the accuracy obtained by the modified AlexNet model (94.68%).

A novel framework entitled EEG-based spatial-temporal CNN (ESTCNN) developed by Gao et al. [14] to detect driver fatigue from EEG signals. The ESTCNN contains two main procedures: a Core Block to deal with the information on the temporal dimension and a dense layer to fuse the spatial features among the electrodes.

The authors implemented a CNN with 14 layers: three core blocks (where each block contains three convolution layers and one max-pooling layer), two dense layers, and a softmax layer. To validate the performance of the proposed framework, the ESTCNN was trained on ten cross-validations for each subject with an average accuracy of 97.37%. In addition, a comparative study of three studies and five competitive models was conducted, and the proposed model demonstrated the highest accuracy of 97.37%.

To solve the drawbacks of the current functional brain network methods (ignore some features of the original EEG signals) and the preprocessing methods (filter out the most noises from signals), Lin et al. [17] proposed three-part architecture for identifying driver drowsiness based on EEG signals. The first part, called front-end CNN is responsible for denoising the raw EEG signal. The second part contains a brain network construction method used to increase the connectivity of EEG channels on a fixed functional brain network with less redundancy. The last part, called the back-end graph neural network, is the fatigue driving recognition model. The proposed framework achieved the highest recognition accuracy of 98.98% compared to commonly used classifiers. In addition, it demonstrated the ability to maintain an accuracy greater than 95% when many channels are affected by noise.

Ko et al. [43] proposed an EEG-based driver drowsiness detection system using differential entropy (DE) with a novel deep convolutional neural network named VIGNet. The raw EEG data features were extracted using the DE method, a conventional machine learning-based method for feature extraction. The extracted features are the inputs to a CNN model with three convolutional layers designed to extract deep and hierarchical features and a dense layer that maps the extracted features to the decision layer. Experiments were conducted on the publicly accessible SEED-VIG dataset. The accuracy of the VIGNet model was 96 %, outperforming the accuracy of the SVM and ESTCNN models.

Chen et al. [47] have used in this study a convolutional neural network (ConvNets) to detect driver drowsiness using raw Multi-Channel EEG signals without using any extraction or selection methods. The proposed architecture consists of 12 layers: five convolutional layers, three max-pooling layers, one mean-pooling layer to extract discriminative features, and three fully-connected layers that optimize the classification process (end-to-end manner). In addition, the authors used a data augmentation strategy to prevent overfitting. To evaluate the used model, the authors used 10- fold cross-validation. based on the obtained, ConvNets performed well compared to other state-of-the-art systems, achieving an accuracy of 97.02 % and a precision of 96.74 %.

Zeng et al. [49] used two classification models, the EEG convolutional (EEGConv) and the EEG convolutional residual (EEG-Conv-R), to classify drivers' mental states using raw EEG signals. Data were collected from ten subjects using 16 channels. The EEGConv architecture contains eight layers: the input layer, three convolutional layers, a pooling layer, an LRN (Local Response Normalization) layer, a fully connected layer, and the output layer. The EEG-Conv-R architecture combines the EEG-Conv with two residual blocks. The two models' evaluation was tested using intra- and inter-subject. EEGConv and EEG-Conv-R achieved greater accuracy than LSTM and SVM models, with 91.788 and 92.682 % accuracy using intra-subjects and 82.95 % and 84.38% accuracy using inter-subjects, respectively. However, the EEG-Conv-R converges faster than the EEGConv.

Table 2: Summarized studies of driver drowsiness detection based-EEG using CNNs

Authors	***Techniques***	***Type of channels***	***Classification Results***
Chaabene et al [3]	• 7 channels. • 7 Layer CNN Model. • Data augmentation.	Multi-Channel	Best classification accuracy: ***90.41%***
Balam et al [6]	• Pz-Oz. • Combination of two CNN models. • Subject-wise, cross-subject-wise, and combined-subjects-wise validations.	Single-Channel	Best classification accuracy: ***combined subjects validation at 94%.***
Ding et al [15]	• Dimension Reduced Level, Feature Extract Level, and Full Connect Level. • 15 Layer Cascaded CNN with an attention mechanism layer.	Single-Channel	Best classification accuracy and Recall: ***97.26% and 96.56% respectively.***

Zhu et al [16]	• CNN with an Inception module. • Modified AlexNet.	Multi-Channel	*Best classification accuracy :* ***95.69% with Inception and 94.68% with AlexNet.***
Lin et al [17]	• Front-end CNN for denoising the raw EEG signal. • Brain network construction method. • Back-end graph neural network.	Multi-Channel	*Best classification accuracy :* ***98.98%***
Gao et al [14]	• EEG-based spatial–temporal CNN (ESTCNN). • 3 Core Blocks, two Dense layers, and a Softmax layer. • Ten cross-validations.	Multi-Channel	*Best classification accuracy :* ***97.37%***
Ko et al [43]	• SEED-VIG dataset. • DE method. • VIGNet Deep Learning Model.	Multi-Channel	*Best classification accuracy :* ***96%***
Zeng et al [49]	• Ten subjects and 16 channels. • The EEGConv and EEG-Conv-R architectures. • Intra- and inter-subject evaluation methods.	Multi-Channel	*Best classification accuracy :* ***91.78% for EEG-Conv and 92.68% EEG-Conv-R*** **using intra-subjects**
Chen et al [47]	• ConvNets with 12 Layers. • Data augmentation strategy. • The 10- fold cross-validation.	Multi-Channel	*Best classification accuracy and precision:* ***97.02% and 96.74 %.***
Houshmand et al [5]	• 1D CNN and two 2D CNN with different parameters. • Dropout activation function. • Raw EEG data, power spectrum analysis EEG data CWT of EEG epochs.	Single-Channel	*Best classification scores:* ***accuracy 94%, recall 95%, precision 91%, and F1-score 93% with CWT-CNN.***
Shalash [7]	• The P4 channel • AlexNet CNN model • Short-time Fourier Transform (STFT). • a modified AlexNet CNN model. • seven different channels (FP1, FP2, T3, T4, O1, O2, and Oz).	Single-Channel	*Best classification accuracy:* ***91% with The T3 channel.***
Guarda et al [44]	• CNN • ULg Multimodality Drowsiness Database. • Fz and Pz channels. • Gray-scale EEG spectrograms. • Seven Layers CNN.	Single-Channel	*Best classification accuracy :* ***86.74%***
Cheng et al	• Raw EEG signal.	Multi-	

[18]	• 256-point FFT. • Basic CNN model. • Leave-One-Subject-Out Cross-Validation.	Channel	*Best classification accuracy :* ***71.15%***
Gao et al [48]	• RN-CNN. • 10 subjects and 30 channels.	Multi-Channel	*Best classification accuracy :* ***92.25%***

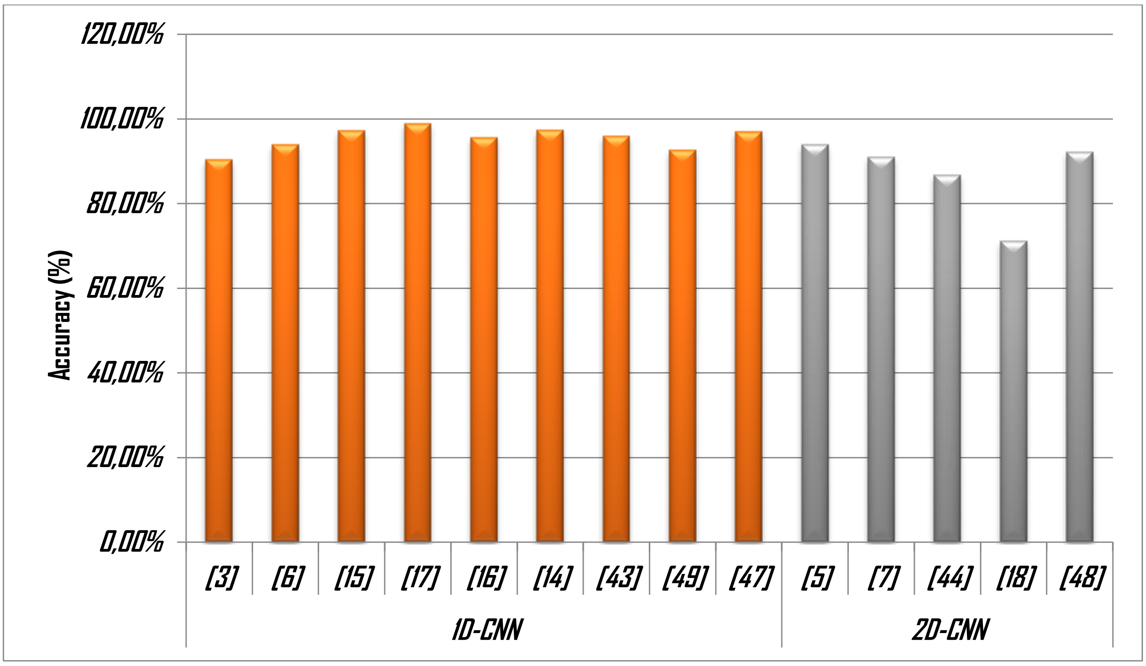


Figure 10: The obtained accuracy (%) by various study using 1D-CNN and 2D-CNN

Budak et al. [19] developed a hybrid model for detecting drowsiness using EEG signals. The developed model utilizes the majority ensemble model to combine three distinct models. Three different groups of features are extracted. The first block extracts frequency, energy, entropy, and rate distribution. The second block extracts Statistical features. In the third block, deep features are extracted from EEG spectrogram images using AlexNet and VGG16. The features extracted from each block are fed into an LSTM network classification model. The result is three different models fused by the majority ensemble model to form the primary model. The authors used the MIT-BIH Polysomnographic database and ten-fold cross-validation to evaluate the performance of the constructed primary model. The obtained average accuracy is 94.31%.

Lee et al. [45] employed a deep neural network with four LSTM layers to classify driver drowsiness based on EEG signals and identify the optimal electrodes. They utilize three classification classes: awakeness, drowsiness, and sleep. In this experiment, 18 EEG channels were used and categorized into eleven groups based on Frontopolar (FP), Dorsolateral Prefrontal Cortex (DLPFC), and Premotor Cortex (PMC). The results indicate that the most accurate group was the FP &DLPFC group, which utilized nearly all channels. Its accuracy was 82.8 %. In this study, the authors observed that the accuracy of awakeness was greater than that of drowsiness across all channel sets, leading them to conclude that the proposed model classified drowsiness data as sleep more frequently than awakeness.

Khessiba et al. [32] proposed two deep learning architectures for detecting drowsiness states in drivers using single-channel EEG signals (Pz-Oz). The proposed models are the 1D-UNet model, designed only with deep 1D-CNN layers, and 1D-UNet-long short-term memory (1D-UNet-LSTM). They were applied to spectral band energy features captured with FFT. The performance of the proposed models is better than other shallow and deep architectures such as the Learning Vector Quantization LVQ, MLP, LSTM, and 1D-CNN-LSTM. The obtained accuracies are 79.3% and 79.4% for 1D-UNet and 1D-UNet-LSTM, respectively, using the ReLU activation function. However, when the SPOCU is used as the activation function, the accuracies are 82% for 1D-UNet and 84% for 1D-UNet-LSTM. The authors implemented the proposed DL models on the RPi 3 device to obtain a real-world evaluation of the proposed drowsiness system. The results indicate that the proposed system slightly increases execution time while maintaining high performance.

Turkoglu et al. [46] proposed a novel hybrid model consisting of deep rhythm features and an LSTM network for EEG-based driver drowsiness detection. The STFT method converts raw EEG signals into time-frequency EEG images. First, five different rhythm images are extracted from the EEG images and fed to CNN pre-trained models ResNet 18, ResNet 50, and ResNet 101 to extract deep features. Next, the extracted features are fed into LSTM layers connected and followed by a fully connected layer, softmax layer, and classification layer to classify whether a driver is in drowsiness or awake state. Two experiments were applied to evaluate the performance of the proposed scheme. The first step involves feeding the EEG images to the features extraction phase without extracting the rhythms images. In this phase, the ResNet 18, ResNet 50, and ResNet 101 were used as feature extractors, while the CNN and SVM were used as classifiers. According to the results, ResNet 18+CNN achieved the highest accuracy of 84.78 %. The second experiment consists of evaluating the proposed rhythm-based-deep features and LSTM networks by changing each time the features extractor. The best result was 97.92% obtained from the three-way concatenation of the ResNet18, ResNet50, and ResNet101 models.

Jeong et al. [50] developed Deep Spatio-Temporal Convolutional Bidirectional LSTM Network (DSTCLN) model to classify pilots' mental states from EEG signals. Based on KSS values, the authors utilized two classes (awakeness, drowsiness) and five classes known as drowsiness classes (very alert (VA), fairly alert (FA), neither alert nor sleepy (NAS), sleepy but making no effort to stay awake (SNEA), and very sleepy (VS)). The data is preprocessed after collecting EEG signals from a simulation environment using 30 EEG channels. The authors used five convolutional blocks of a Spatio-Temporal CNN to extract high-level Spatio-temporal features. The extracted features were fed into a Bi-LSTM with four Bi-LSTM layers, and a dropout layer was used to reflect the temporal information of time-series data using. The classification layer consists of three fully connected layers and a softmax layer. The deep model achieved an accuracy of 87% for 2-class and 69% for 5-class. Comparing the proposed model to other conventional techniques revealed that the DSTCLN achieved the best classification performance.

Michielli et al. [51] developed a novel cascaded RNN architecture based on long short-term memory (LSTM) for classifying sleep stages based on Single-Channel EEG signals. They proposed two RNNs-based LSTM models common in the three first steps: data acquisition, signal preprocessing, and feature extraction. Fifty-five features were extracted (time domain and frequency domain). In the selection process, which reduces the computation cost and selects the most relevant features, the minimum redundancy maximum relevance (mRMR) was utilized in the first model. In contrast, the Dimensionality reduction (PCA) has been used in the second model. In the classification step, the first model inputs are the outputs of the mRMR to classify 4-class, and the second network uses the outputs of the PCA method to classify 2-class; finally, the two models were connected using a cascaded architecture to classify five sleep stages. The cascaded RNN architecture achieved an average accuracy of 86.7%.

Table 3: Summarized studies of driver drowsiness detection based-EEG using RNNs

Authors	***Techniques***	***Type of channels***	***Classification Results***
Budak et al [19]	• Hybrid model. • Majority ensemble model. • Three groups of features. • EEG spectrograms • LSTM network model. • MIT-BIH database. • Ten-fold cross-validation.	Multi-Channel	*Best classification accuracy:* ***94.31%***
Lee et al [45]	• Identify the optimal electrodes.	Multi-Channel	*Best classification*

	• Three classification classes. • 18 EEG channels. • LSTM network model.		*accuracy:* ***82.8%***
Khessiba et al [32]	• Pz-Oz Channel. • 1D-UNet-LSTM. • Spectral band energy features captured with FFT. • RPi 3 device for a real-world evaluation.	Single-Channel	*Best classification accuracy:* ***84%***
Turkoglu et al [46]	• STFT time-frequency EEG images. • ResNet 18, ResNet 50, and ResNet 101 as extractors. • LSTM classification model. • ResNet+LSTM model.	Multi-Channel	*Best classification accuracy:* ***97.92%***
Jeong et al [50]	• 30 EEG channels. • Spatio-Temporal CNN. • Bi-LSTM model. • DSTCLN	Multi-Channel	*Best classification accuracy:* ***87% (2-class) 69% (5-class)***
Michielli et al [51]	• Cascaded RNN • Fifty-five features. • (mRMR) and (PCA).	Single-Channel	86.7%

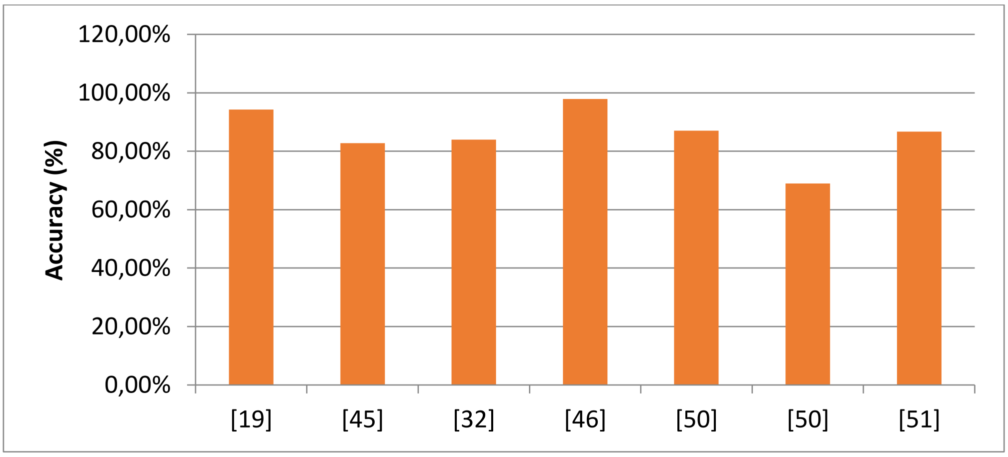


Figure 11: The obtained accuracy (%) by various study using RNNs

4 Discussion

Detecting drowsiness has been and remains an essential task because it affects the performance and throughput of persons and can lead to various negative outcomes, such as car accidents and crashes. EEG signals are the gold standard for monitoring drowsiness. In the past, many researchers have focused their studies on which brain regions are most informative in detecting drowsiness (see [1]). Most studies found that frontal, parietal, and occipital regions are the most informative [1]. In addition, some researchers oriented their research on which are the informative electrodes to reduce electrodes numbers. Researchers recently adopted artificial intelligence techniques to classify whether a driver is in a drowsy or awake state such as machine learning methods, which gave good results [10, 53, 54, and 55]. However, they required features extraction (hand-crafted extraction) and features selection (optional) steps that influence the results. Due to the existing of many methods and techniques in this area, they also need a massive amount of data. According to these limitations in the last years, most studies have used deep learning techniques such as CNN and its variants, RNNs, and its variants to detect drowsiness based on EEG signals. DL techniques have shown notable results in processing biomedical signals.

Our analysis shows that the most recent driver drowsiness-based deep learning works have used the convolutional neural network (CNN). The latter showed efficient performances in signals classification. The results in Table 2 have proved this where all the obtained accuracies exceed 90% except on [18], which was 71.15%, which is a good result considering the use of raw EEG signal without using any artifact removal methods. RNN networks also have been used as they perform well in time-series and sequential data; the results shown in Table 3 demonstrate that they are good candidates for this aim, where the highest accuracy was 97.92 % [46] and the lowest was 82.8% [45]. However, it isn't easy to compare the reviewed works due to the use of different datasets, types of channels, types of inputs, and classification model architectures. If we take the [3] and [16] works, both have used multi-channel EEG signals and EEG signals as data and CNN as a classifier. Still, they have got different results because they used different CNN architectures where the model of [3] contains four convolutional layers, one max-pooling layer, and two fully connected layers. However, in [16], it contains five convolutional layers, two pooling layers, three Inception modules, and three fully connected layers.

In addition, we have observed that in recent years, researchers in the driver drowsiness-based EEG field have shifted their focus toward using Single-Channel EEG recordings rather than Multi-Channel [6, 15, 5, 32, 51, and 7]. The main reason for this redirection is that the Multi-Channel EEG signals require a larger storage capacity and high computing time and are more expensive than single-channel records. We cannot compare the obtained results in terms of single or multi-channel. Each work employs different wearable devices, preprocessing techniques, and models. However we can say that single-channel research papers have yielded prominent results (see Tables 2 and 3). Perhaps in the future, they will be more accurate.

Another point to discuss is that in some works, instead of using EEG as signals, the authors have used some methods such as the continuous wavelet transform (CWT) [5] and Short-time Fourier Transform (STFT) [7]; to convert EEG signals to time-frequency domain images (2D spectrogram). The 2D spectrograms are then fed to CNN model that demonstrated high image classification and pattern detection performance. For example, in [5], three models were developed 1D CNN with raw EEG data, 2D CNN with power spectrum analysis EEG data and CWT-CNN with 2D spectrogram; the highest accuracy was 91% obtained by CWT-CNN. The results of [5] and [7] are encouraging, and we believe that using power preprocessing methods and the CNN model with spectrogram images will be much better.

After reading many papers on driver drowsiness detection and especially on deep learning for EEG-based driver drowsiness detection, we have noticed that all works' main problem is developing a higher accuracy system. However, we need an accurate system that can detect a drowsy state in a short time and requires small spatial memory and few computational resources. A short time because the early detection helps to avoid accidents. Memory space and computational resources must be reduced because the device where the model is integrated will be in the vehicle, which is usually a smartphone. A smartphone with an enormous capacity costs a lot.

5 Challenges and future works

In conclusion, EEG sensors are useful for detecting weariness and drowsiness using DL techniques. However, various challenges and constraints persist today, impeding the development of real-world applications. The first one is the lack of data. Most of the studies have used a few participants (50 participants or less), which can influence the results of the proposed models. As it is common knowledge, DL models require enormous data for training. In addition, the datasets should be diversified so that the models can be general, efficient, and robust. The second challenge is comparing model performance with other states of the art, as in most papers, to validate that the proposed model is better than others. However, this comparison is not reliable and unfair, as each study uses different datasets collected under different experimental conditions, the number of electrodes, sampling frequency, and the number of participants. Therefore, to obtain a fair comparison, it is recommended that all models utilize the same dataset, which is challenging due to the use of private datasets.

The third challenge is the need for powerful preprocessing methods and techniques to remove artifacts and unwanted signals from the original EEG signals without information loss. The raw EEG signal (original) is affected by various noises, such as eye blinking and muscle noises, which decrease its quality and, as a result, affect the detection model's performance. So, a preprocessing step is required to clean the signals and improve their quality.

Deep layer models with massive data may give high performance and accuracy but require high computational resources. Therefore, the last challenge is the need for powerful hardware to implement, train, and store Deep-layer models with massive data.

The fifth challenge is that most researchers typically use virtual or simulated environments to conduct their studies and develop their final system outcomes. Nevertheless, it is important to note that these results may not accurately reflect actual driving conditions, which impacts the system's reported accuracy.

The sixth challenge is that the driver may be uncomfortable because of the equipment and sensors attached to his body. In addition, even minor motion can introduce noise into the extracted signals, diminishing their precision.

For future studies, we recommend:

- Using data augmentation techniques and strategies to overcome the lack of data, improve accuracy, and achieve an acceptable generalization.

- Use available online datasets to make a reliable comparison of models, and choose the most powerful one for detecting drowsiness instead of using private datasets.

- Combine the EEG with other physiological signals such as EOG to get more accurate and efficient results.

- Implementing deeper models (with more layers) to automatically learn and extract the most prominent EEG features.

- The use of single-channel EEG signals that require low storage capacity, low computing time, and cost less than the multi-channel record, with CNN, to achieve prominent results.

- Use the cloud to avoid the necessity of powerful hardware to train it with massive data.

- Verify the accuracy of the system's findings by conducting an actual driving situation.

- After determining that the driver is experiencing drowsiness, the system should promptly notify the driver or any close traffic patrol about their possible loss of focus by generating noise or causing the steering wheel or seat to vibrate. The system may additionally prompt the driver to pause and rest, particularly if they have been driving for a prolonged duration, or enable a smooth transition to autonomous driving mode.

- Utilizing sensors to monitor the road ahead and detect probable collisions to prevent accidents caused by proximity with other cars. Once the system detects an impending accident, it can autonomously engage the brakes or issue visual and auditory alerts. This action mitigates the impact's severity or entirely averts the collision.

6 Conclusion

This paper has reviewed novel research papers on detecting driver drowsiness or fatigue using EEG and deep learning techniques such as CNNs and RNNs models. Efficient results were obtained from both DL instances, where the higher accuracy is 98.98%, and the low accuracy is 71.15% without using any artifacts removals, which is an acceptable score. In addition, we have discussed the reasons for utilizing single-channel EEG signals rather than multi-channel EEG signals in certain works and the reasons for using 2D spectrogram EEG images rather than EEG signals. At last, we have focused on some limitations of the proposed systems. For example, the importance of considering the time, accuracy, and costs (reducing the spatial memory and computational resources), the lack of data, and others are mentioned as challenges.

Declarations

Conflict of interest

The author declares that they have no conflict of interest.

Data availability statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [Imene Latreche], [Sihem Slatnia], [Okba Kazar], [Saad Harous] and [Ezedin Barka]. The first draft of the manuscript was written by [Imene Latreche] and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

[1] Majumder, S., Guragain, B., Wang, C., & Wilson, N (2019, May) On-board drowsiness detection using EEG: Current status and future prospects. In *2019 IEEE International Conference on Electro Information Technology (EIT)* (pp. 483-490). IEEE. https://doi.org/10.1109/eit.2019.8833866

[2] Dutta, A., Kour, S., & Taran, S (2020) Automatic drowsiness detection using electroencephalogram signal. *Electronics Letters, 56*(25), 1383-1386.

https://doi.org/10.1049/el.2020.2697

[3] Chaabene, S.; Bouaziz, B.; Boudaya, A.; Hökelmann, A.; Ammar, A.; Chaari, L (2021) Convolutional Neural Network for Drowsiness Detection Using EEG Signals. Sensors 2021, 21,1734. https://doi.org/10.3390/s21051734

[4] Covington, T (2021, September 30) *Drowsy Driving Statistics*. THE zebra. https://www.thezebra.com/resources/research/drowsy-driving-statistics/?__cf_chl_captcha_tk__=hzpmyRUeMjxexnLGFixSOgDVfrH_5vqfUbB0LwVfJIM-1637260303-0-gaNycGzNBiU#statistics-2020 . Accessed March 2022

[5] Houshmand, S., Kazemi, R., & Salmanzadeh, H (2021) A novel convolutional neural network method for subject-independent driver drowsiness

detection based on single-channel data and EEG alpha spindles. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of engineering in medicine, 09544119211017813. https://doi.org/10.1177/09544119211017813

[6] Balam, V. P., Sameer, V. U., & Chinara, S (2021) Automated classification system for drowsiness detection using convolutional neural network and electroencephalogram. IET Intelligent Transport Systems, 15(4), 514-524. https://doi.org/10.1049/itr2.12041

[7] Shalash, W. M (2019, December) Driver fatigue detection with single EEG channel using transfer learning. In 2019 IEEE International Conference on Imaging Systems and Techniques (IST) (pp. 1-6). IEEE. https://doi.org/10.1109/ist48021.2019.9010483

[8] Doma, V., & Pirouz, M (2020) A comparative analysis of machine learning methods for emotion recognition using EEG and peripheral physiological signals. *Journal of Big Data*, *7*(1), 1-21. https://doi.org/10.1186/s40537-020-00289-7

[9] Gwak, J., Shino, M., & Hirao, A (2018, November) Early detection of driver drowsiness utilizing machine learning based on physiological signals, behavioral measures, and driving performance. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1794-1800). IEEE. https://doi.org/10.1109/itsc.2018.8569493

[10] Hu, J., & Min, J (2018) Automated detection of driver fatigue based on EEG signals using gradient boosting decision tree model. *Cognitive neurodynamics*, *12*(4), 431-440. https://doi.org/10.1007/s11571-018-9485-1

[11] Mu, Z., Hu, J., & Yin, J (2017) Driving fatigue detecting based on EEG signals of forehead area. *International Journal of Pattern Recognition and Artificial Intelligence*, *31*(05), 1750011. https://doi.org/10.1142/s0218001417500112

[12] Ma, Y., Zhang, S., Qi, D., Luo, Z., Li, R., Potter, T., & Zhang, Y (2020) Driving drowsiness detection with EEG using a modified hierarchical extreme learning machine algorithm with particle swarm optimization: A pilot study. *Electronics*, *9*(5), 775. https://doi.org/10.3390/electronics9050775

[13] Zeng, H., Yang, C., Dai, G., Qin, F., Zhang, J., & Kong, W (2018) EEG classification of driver mental states by deep learning. *Cognitive neurodynamics*, *12*(6), 597-606. https://doi.org/10.1007/s11571-018-9496-y

[14] Gao, Z., Wang, X., Yang, Y., Mu, C., Cai, Q., Dang, W., & Zuo, S (2019) EEG-based spatio–temporal convolutional neural network for driver fatigue evaluation. IEEE transactions on neural networks and learning systems, 30(9), 2755-2763. https://doi.org/10.1109/tnnls.2018.2886414

[15] Ding, S., Yuan, Z., An, P., Xue, G., Sun, W., & Zhao, J (2019, November) Cascaded convolutional neural network with attention mechanism for mobile eeg-based driver drowsiness detection system. In 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 1457-1464). IEEE. https://doi.org/10.1109/bibm47256.2019.8982938

[16] Zhu, M., Chen, J., Li, H., Liang, F., Han, L., & Zhang, Z (2021) Vehicle driver drowsiness detection method using wearable EEG based on convolution neural network. Neural computing and applications, 1-16. https://doi.org/10.1007/s00521-021-06038-y

[17] Lin, Z., Qiu, T., Liu, P., Zhang, L., Zhang, S., & Mu, Z. (2021). Fatigue driving recognition based on deep learning and graph neural network. Biomedical Signal Processing and Control, 68, 102598. https://doi.org/10.1016/j.bspc.2021.102598

[18] Cheng, E. J., Young, K. Y., & Lin, C. T (2018, November) Image-based EEG signal processing for driving fatigue prediction. In 2018 International Automatic Control Conference (CACS) (pp. 1-5). IEEE. https://doi.org/10.1109/cacs.2018.8606734

[19] Budak, U., Bajaj, V., Akbulut, Y., Atila, O., & Sengur, A (2019) An effective hybrid model for EEG-based drowsiness detection. IEEE sensors journal, 19(17), 7624-7631. https://doi.org/10.1109/jsen.2019.2917850

[20] Majumder, S., Guragain, B., Wang, C., & Wilson, N (2019, May) On-board drowsiness detection using EEG: Current status and future prospects. In *2019 IEEE International Conference on Electro Information Technology (EIT)* (pp. 483-490). IEEE. https://doi.org/10.1109/eit.2019.8833866

[21] Ma, J., Gu, J., Jia, H., Yao, Z., & Chang, R (2018) The relationship between drivers' cognitive fatigue and speed variability during monotonous daytime driving. *Frontiers in psychology*, *9*, 459. https://doi.org/10.3389/fpsyg.2018.00459

[22] Im C-H, I. (2018). Computational EEG Analysis: Methods and Applications. Im C.-H., editor. https://doi.org/10.1007/978-981-13-0908-3

[23] Shoeibi, A., Khodatars, M., Ghassemi, N., Jafari, M., Moridian, P., Alizadehsani, R., ... & Acharya, U. R (2021) Epileptic Seizures Detection Using Deep Learning Techniques: A Review. *International Journal of Environmental Research and Public Health*, *18*(11), 5780. https://doi.org/10.3390/ijerph18115780

[24] Houssam M (28 septembre 2020) Deep Learning: What is it? DataScientest. https://datascientest.com/deep-learning-definition Accessed March 2021

[25] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L (2021) Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*(1), 1-74. https://doi.org/10.1186/s40537-021-00444-8

[26] Al-Saegh, A., Dawwd, S. A., & Abdul-Jabbar, J. M (2021) Deep learning for motor imagery EEG-based classification: A review. *Biomedical Signal Processing and Control*, *63*, 102172. https://doi.org/10.1016/j.bspc.2020.102172

[27] Sarker, I. H (2021) Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 1-20. https://doi.org/10.1007/s42979-021-00815-1

[28] Mao, W. L., Fathurrahman, H. I. K., Lee, Y., & Chang, T. W (2020) EEG dataset classification using CNN method. In *Journal of physics: conference series* (Vol. 1456, No. 1, p. 012017). IOP Publishing. https://doi.org/10.1088/1742-6596/1456/1/012017

[29] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J (2021) 1D convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, *151*, 107398. https://doi.org/10.1016/j.ymssp.2020.107398

[30] Apostolopoulos, I. D., & Mpesiana, T. A (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine, 43(2), 635-640. https://doi.org/10.1007/s13246-020-00865-4

[31] Hussain, M., Bird, J. J., & Faria, D. R (2018, September) A study on cnn transfer learning for image classification. In *UK Workshop on computational Intelligence* (pp. 191-202). Springer, Cham. https://doi.org/10.1007/978-3-319-97982-3_16

[32] Khessiba, S., Blaiech, A. G., Khalifa, K. B., Abdallah, A. B., & Bedoui, M. H (2020) Innovative deep learning models for EEG-based vigilance detection. *Neural Computing and Applications*, 1-17. https://doi.org/10.1007/s00521-020-05467-5

[33] Aleia Knight (2021, April) LSTM Neural Network: The Basic Concept. Towards data science. https://towardsdatascience.com/lstm-neural-network-the-basic-concept-a9ba225616f7 Accessed March 2021

[34] Jason Brownlee (May, 2017) A Gentle Introduction to Long Short-Term Memory Networks by the Experts. Machine learning mastery. https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/ Accessed March 2021

[35] Ambika Choudhury (October, 2020) Gated Recurrent Unit – What Is It And How To Learn. Analytics in diamag. https://analyticsindiamag.com/gated-recurrent-unit-what-is-it-and-how-to-learn/ Accessed March 2021

[36] B Kemp, AH Zwinderman, B Tuk, HAC Kamphuisen, JJL Oberyé (2000) Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. IEEE-BME 47(9):1185-1194. https://doi.org/10.1109/10.867928

[37] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E (2000) PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220. https://doi.org/10.1161/01.cir.101.23.e215

[38] Min, Jianliang; Wang, Ping; Hu, Jianfeng (2017) The original EEG data for driver fatigue detection. figshare. Dataset. https://doi.org/10.6084/m9.figshare.5202739.v1 Accessed March 2021

[39] Cao, Zehong; Chuang, Michael; King, J.T.; Lin, Chin-Teng (2019) Multi-channel EEG recordings during a sustained-attention driving task (pre-processed dataset). figshare. Dataset. https://doi.org/10.6084/m9.figshare.7666055.v3 Accessed March 2021

[40] Y. Ichimaru and G. B. Moody (1999) Development of the Polysomnographic database on CD-ROM," *Psychiatry Clin. Neurosci.*, vol. 53, no. 2, pp. 175–177, Apr. 1999. https://doi.org/10.1046/j.1440-1819.1999.00527.x

[41] Zheng, W. L., & Lu, B. L (2017) A multimodal approach to estimating vigilance using EEG and forehead EOG. *Journal of neural engineering*, *14*(2), 026017. https://doi.org/10.1088/1741-2552/aa5a98

[42] J. Min, P. Wang, and J. Hu 2017 Driver fatigue detection through multiple entropy fusion analysis in an EEG-based system," *PLoS One*, vol. 12, no. 12, Dec. 2017. https://doi.org/10.1371/journal.pone.0188756

[43] Ko, W., Oh, K., Jeon, E., & Suk, H. I (2020, February) Vignet: A deep convolutional neural network for eeg-based driver vigilance estimation. In 2020 8th International Winter Conference on Brain-Computer Interface (BCI) (pp. 1-3). IEEE. https://doi.org/10.1109/bci48061.2020.9061668

[44] Guarda, L., Astorga, N., Droguett, E., Moura, M., & Ramos, M (2018) Drowsiness detection using electroencephalography signals: A deep learning based model. Proceedings of the Probabilistic Safety Assessment and Management PSAM, Los Angeles, CA, USA, 16. https://doi.org/10.1016/j.eswa.2022.116977

[45] Lee, C., Choi, R. H., & An, J (2021, February) Deep Neural Network for Drowsiness Detection from EEG. In 2021 9th International Winter Conference on Brain-Computer Interface (BCI) (pp. 1-3). IEEE. https://doi.org/10.1109/bci51272.2021.9385368

[46] Turkoglu, M., Alcin, O. F., Aslan, M., Al-Zebari, A., & Sengur, A (2021) Deep rhythm and long short term memory-based drowsiness detection. Biomedical Signal Processing and Control, 65, 102364. https://doi.org/10.1016/j.bspc.2020.102364

[47] Chen, J., Wang, S., He, E., Wang, H., & Wang, L (2021) Recognizing drowsiness in young men during real driving based on electroencephalography using an end-to-end deep learning approach. Biomedical Signal Processing and Control, 69, 102792. https://doi.org/10.1016/j.bspc.2021.102792

[48] Gao, Z. K., Li, Y. L., Yang, Y. X., & Ma, C (2019) A recurrence network-based convolutional neural network for fatigue driving detection from EEG. Chaos: An Interdisciplinary Journal of Nonlinear Science, 29(11), 113126. https://doi.org/10.1063/1.5120538

[49] Zeng, H., Yang, C., Dai, G., Qin, F., Zhang, J., & Kong, W (2018) EEG classification of driver mental states by deep learning. Cognitive neurodynamics, 12(6), 597-606. https://doi.org/10.1007/s11571-018-9496-y

[50] Jeong, J. H., Yu, B. W., Lee, D. H., & Lee, S. W (2019) Classification of drowsiness levels based on a deep spatio-temporal convolutional bidirectional LSTM network using electroencephalography signals. Brain sciences, 9(12), 348. https://doi.org/10.3390/brainsci9120348

[51] Michielli, N., Acharya, U. R., & Molinari, F (2019) Cascaded LSTM recurrent neural network for automated sleep stage classification using single-channel EEG signals. Computers in biology and medicine, 106, 71-81. https://doi.org/10.1016/j.compbiomed.2019.01.013

[52] Latreche, I., Slatnia, S., & Kazar, O. (2022, October). CNN-LSTM To Identify The Most Informative EEG-Based Driver Drowsiness Detection Brain Region. In *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 725-730). IEEE. https://doi.org/10.1109/ismsit56059.2022.9932696

[53] H. Shabani, M. Mikaili, and S. M. R. Noori (2016) Assessment of recurrence quantification analysis (rqa) of eeg for development of a novel drowsiness detection system. *Biomed. Eng. Lett.*, Vol. 6, no. 3, pp. 196–204. https://doi.org/10.1007/s13534-016-0223-5

[54] A. Jalilifard, and E. B. Pizzolato (2016) An efficient k-nn approach for automatic drowsiness detection using singlechannel eeg recording. in 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, pp.820–824. https://doi.org/10.1109/embc.2016.7590827

[55] G. Li, andW.-Y. Chung (2015) A contextaware eeg headset system for early detection of driver drowsiness. *Sensors*, Vol. 15, no. 8, pp. 20873–20893. https://doi.org/10.3390/s150820873

[56] Albadawi, Y., Takruri, M., & Awad, M. (2022). A review of recent developments in driver drowsiness detection systems. *Sensors*, *22*(5), 2069. https://doi.org/10.3390/s22052069

Inverse Fuzzy Fault Models for Fault Isolation and Severity Estimation in Industrial Pneumatic Valves

M.F. Ávila-Díaz[1], M.A. Márquez-Vera[1], O. Díaz-Parra[1], V. Puig[2], A. Ma'arif[3]
[1]Polytechnic University of Pachuca, Carr. Pachuca-Cd. Sahagún Km 20, Zempoala 43830, Hidalgo, Mexico
[2]Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial, Llorens i Artigas, 4-6, 08028 Barcelona, Spain
[3]Universitas Ahmad Dahlan, Jl. Kapas 9, Semaki, Kec. Umbulharjo, Yogyakarta 55166, Indonesia
E-mail: marquez@upp.edu.mx

Keywords: fault diagnosis, pneumatic valves, fuzzy modeling, actuator fault

Received: August 17, 2023

Fault detection is crucial in the chemical industry for identifying process problems, and determining the nature of the fault is essential for scheduling maintenance. This study focuses on the application of inverse fuzzy models to reconstruct faults for the purpose of detection, isolation, and classification. By inverting fuzzy models, the fault signal can be reconstructed, enabling identification of the fault source and its characteristics. To address the issue of undetected small abrupt faults, we employed the wavelet transform. This approach allows for the detection of incipient faults, while the classification is achieved by evaluating the response of the fault reconstruction. Fault isolation is accomplished by comparing the reconstructed faults. However, in the case of the pneumatic valve utilized, four out of the 19 simulated faults demonstrated poor isolation due to the similarity of their reconstructions using inverse fuzzy models. We also present a comparison with similar applications in existing literature. The fault detection rate obtained in this study is 84.81%, which is higher compared to the rates of 55.45% and 82.37% reported in other works. Additionally, the accuracy achieved in this work is 78.85%, indicating the ratio of correctly classified faults to the total number of measurements, including both fault and no-fault conditions.

Povzetek: V članku je predstavljen model za zaznavanje, izolacijo in oceno napak v industrijskih pnevmatskih ventilih, ki dosega visoko uspešnost zaznavanja napak.

1 Introduction

Ensuring safety is of paramount importance in the chemical industry. Various processes within this industry are susceptible to faults, which not only pose a risk to product integrity but also endanger the safety of factory operators [1]. Startling statistics from Japan reveal that over the past five years, approximately one-fourth of chemical industries reported more than 20 accidents [2]. In a comprehensive analysis of 170 accidents, [3] found that 56.2% of these cases involved explosions, including a notable incident on August 12, 2015, in Tianjin, China, which resulted in 165 fatalities and eight missing individuals. Hence, the ability to detect faults assumes paramount importance in the chemical industry [4]. Recognizing the limitations of the defense-in-depth approach in averting accidents like the Fukushima Daiichi nuclear disaster in 2011, [5] proposed implementing the safety diagnosability principle to enhance situational awareness. This approach facilitates the development of maintenance schedules for optimal operational performance and process safety [6], while also enabling timely response during emergencies.

A fault refers to an unexpected deviation in a system's behavior. The detection of faults serves as a crucial first step in fault diagnosis, indicating the presence of a problem. Subsequently, fault isolation becomes necessary to identify the specific nature of the issue. This combined process of fault detection and isolation (FDI) is often referred to as fault diagnosis in many research papers [7]. Some authors further include an identification stage to assess the magnitude of the fault. As a result, the comprehensive execution of these stages constitutes the overall fault diagnosis procedure [8].

Two main techniques are commonly employed for FDI applications. The first technique involves utilizing an approximate model to compute the discrepancy between the real process outputs and the model-derived outputs, referred to as the residual signal. The detection of faults is achieved by analyzing the magnitude of the residual signal [9]. Fault isolation, on the other hand, involves examining the fault's characteristics, such as its frequency response or employing symptom mapping from the symptom space to the fault space, as demonstrated by [8]. The second technique for fault isolation involves utilizing measured data from the process to search for discernible patterns. Statistical methods, such as Hotelling's statistic (T^2-chart) and the squared prediction error (Q-chart), have been employed for this purpose [10]. Additionally, fault diagnosis in pneumatic valves has been demonstrated using principal component analysis [11]. Alternatively, soft computing techniques like fuzzy logic and artificial neural networks (ANN) have also been utilized for fault diagnosis [12].

When sensor-measured data is utilized for FDI, the methodology is known as data-driven. This approach leverages past information on the process both with and without faults to detect and classify the presence of faults, primarily employing artificial neural networks (ANNs) [13]. Deep neural networks are particularly effective in extracting features that enhance FDI performance. For instance, in fault diagnosis of Tennessee Eastman processes, [14] employed a combination of convolutional ANNs and long short-term memory units. Similarly, [15] implemented an adaptive convolutional ANN for multiscale feature extraction in the same process.

Additionally, there are publications where system behavior analysis is employed to identify new operating points or peculiar dynamics indicative of faults, even in the absence of prior information [16]. However, it is important to note that data-driven methods, as pointed out by [17], may struggle with correlating acquired samples with unlabeled data. Moreover, [18] specified that these methods are susceptible to data sparsity and proposed the use of deep temporal clustering to address this challenge.

1.1 Case of study

Electro-pneumatic valves find widespread usage in industries such as chemical [19], biotechnology, food processing [20], cement production [21, 22], and energy generation [23]. These valves play a vital role in regulating fluid flow within processing pipelines. In certain industrial contexts, valves are also referred to as actuators. However, these devices can be susceptible to issues such as erosion and degradation [24, 25], which may lead to faults with potential cascading effects throughout the entire process [26]. Additionally, faults can originate from the spring component used in hydraulic valves [27].

A project called the Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems (DAMADICS) was proposed to develop online diagnostic tools for a pneumatic valve, specifically simulating a sugar evaporation station in Cukrownia Lublin, Poland [9]. The purpose of the process is to maintain the syrup level between 14% and 70% for juice condensation, as a lower value can lead to overheating of the evaporation chamber, while higher values can result in contamination of other stations within the factory [28]. The developed model accurately simulates real phenomena that can potentially cause faults in the valve actuator, making it highly applicable in real-world scenarios. This platform has been utilized to develop fault detection and isolation (FDI) techniques without the need for extensive analytical knowledge of the valve or its associated faults [29].

The DAMADICS model comprises three key components: a control valve, a servomotor, and a positioner. The model incorporates five inputs: i) the control variable C_V, ii) the liquid pressure before the valve P_1, iii) the liquid pressure behind the valve P_2, iv) the liquid temperature T, and v) the faults vector f. The model produces two output signals: the displacement of the valve head X and the liquid flow rate F [28].

In this system, the linear rod motion is determined by the pressure force, which is achieved using a flexible diaphragm. The movement of the rod controls the internal area of the valve, thereby influencing the fluid force, which is dependent on the pressure difference between P_1 and P_2 [9]. A PID controller is responsible for regulating the air within the chamber to maintain a desired set point. The model has the capability to simulate 19 distinct faults, denoted as F_1, F_2, . . ., F_{19}.

1.2 State of the art

Now, let's explore the state-of-the-art research in fault detection and isolation (FDI) for the DAMADICS benchmark, focusing primarily on studies that employ data-driven techniques.

The utilization of residual signals for fault detection is a widely employed approach in FDI. [30] introduced an interval model to derive nonlinear interval observers. These observers are utilized to generate residual signals for fault detection, and subsequently, the responses from a set of observers are compared to fault signatures for fault isolation. Soft computing techniques can also be employed to model nonlinear systems [31]. In model-based FDI, the evaluation of residuals plays a crucial role in fault detection, and an interesting methodology involves the implementation of fuzzy logic.

Fuzzy models were proposed by [9] to approximate the variables X and F by utilizing the input variables. Residual signals were employed for fault detection, followed by the use of 19 fuzzy classifiers to isolate the specific faults.

[32] improved the fuzzy partition of classifiers by incorporating clustering algorithms. Their methodology enables the detection of new operation modes in a model-free manner, meaning that it is data-driven and does not rely on prior fault information. They implemented adaptive fuzzy rules using an incremental unsupervised Gaussian participatory clustering procedure, which shows promise in terms of its ability to adapt without prior knowledge of faults.

Another approach utilizing fuzzy logic for FDI was demonstrated by [8]. In their study, they employed fuzzy generalized nearest prototype classifiers and applied 20 distinct magnitudes of faults. It is noteworthy that they developed a classifier capable of isolating different magnitudes within the same fault.

In addition, [33] utilized artificial neural networks (ANN) for FDI. They employed ANN models to represent the behavior of the DAMADICS benchmark both with and without faults in order to calculate the residuals. By evaluating the magnitude of these residuals, it becomes feasible to detect faults. For the fault isolation stage, ANN models were utilized for each potential fault under evaluation within a specific time window. A larger time window provides higher confidence with a delay, whereas a shorter time window enables early isolation but with lower confidence.

[36] employed an ANN to develop a neural model for the DAMADICS benchmark and calculate residual signals. The fault isolation process involved utilizing a decision tree to identify unique signatures or footprints in the symptoms. This approach offered the advantage of reduced time required for FDI, and it successfully isolated 12 distinct faults.

On the other hand, [33] identified three fault groups that exhibit similar symptoms. Interestingly, even the fault-free scenario can be mistakenly isolated within these groups. Specifically, there exists a group comprising faults F_5, F_8, and F_{14} as potential candidates due to their shared symptom. This similarity poses a challenge for fault isolation.

Neuro-fuzzy systems combine the computational power and learning capability of ANNs with the reasoning and interpretability of fuzzy logic [34]. These systems can be viewed as either fuzzy systems adapted as ANNs or ANNs with fuzzy sets and operators as their units [12]. In the context of the benchmark used in this study, [12] implemented multiple observers for FDI. These observers, known as unknown input observers, were combined using fuzzy logic to generate output estimates. Notably, their approach involved using the Gustafson-Kessel clustering for the initial fuzzy partition and employing genetic programming to obtain state-space subsystems as the rule consequents. Additionally, they proposed the use of one linear model to approximate X and five linear submodels for F. Importantly, they emphasized that faults can have varying effects at different operating points. Their methodology, known as the neuro-fuzzy and decoupling fault diagnosis scheme, showed promising results.

A similar approach was demonstrated by [35], who utilized an ANN model to approximate the process behavior. Subsequently, a neuro-fuzzy classifier was employed to detect and isolate faults based on the residual signals. This method follows a model-based approach, and it successfully detected and isolated two specific faults.

[37] introduced a pattern-recognition approach using radial basis function networks for classification. The results were then aggregated using a fuzzy system to generate a decision signal. In their study, they successfully isolated faults F_1, F_2, F_7, F_8, $F_{10} - F_{13}$, and $F_{15} - F_{19}$ in abrupt scenarios.

Self-organizing maps (SOM), similar to ANN, have been utilized by [38] for fault diagnosis. In their approach, fault detection is achieved by computing the difference between the obtained signals and the healthy behavior of DAMADICS. However, it should be noted that some faults were undetectable using this method. A significant drawback of this approach is the considerable time required for online diagnosis. Another application of SOM for fault classification was demonstrated by [39], where they successfully classified three faults, even in cases where fault classes overlapped.

An insightful overview of computational intelligence techniques applied in FDI was presented by [40]. They explored the application of various techniques such as fuzzy logic, ANNs, neuro-fuzzy systems, and genetic algorithms in FDI for diverse systems, including gas turbines, conductive flow systems, and aero-engines. In particular, Chapter 1 of [41] highlights the significance of FDI in industrial processes and discusses the utilization of computational intelligence techniques to address challenges such as local nonlinearities, noise, and uncertainty.

A distinct approach was introduced by [42], where FDI was tackled by verifying the consistency of the current fault identification with a specific fault set using timed automata. This approach does not necessitate prior information about the system. The faults considered in this study were F_{16}, F_{18}, and F_{19}, and they were successfully isolated after 16 sample times in the DAMADICS simulation. Another application of fault detection was presented by [43], where a methodology called "typically and eccentricity data analysis" was employed for online fault detection without the need for prior knowledge of the system. The obtained results showcased a true positive rate (fault detection rate) of 74.96% for faults $F_{16} - F_{19}$.

Wear and tear on systems can lead to the development of incipient faults, and they can also provide insights into the actuator's remaining lifespan [44]. Detecting incipient faults can be challenging as they may resemble modifications in the operating point. Early detection is crucial for such faults since they exhibit a continuous and gradual development that may only be detectable when their magnitude approaches 50%. [28] employed a hidden Markov model to detect both abrupt faults (sudden appearance) and incipient faults. This data-driven methodology utilized 46 distinct symbols to differentiate between different operating conditions.

In a review conducted by Capaci et al. [29], the authors explored the field of smart diagnosis in control valves, focusing on the analysis of both normal and abnormal operating conditions. The study examined various statistical and soft computing techniques to compare and analyze the results obtained from these conditions. The review aimed to provide insights into the effectiveness of different diagnostic approaches in detecting and characterizing valve abnormalities.

Table 1 lists the characteristics of works mentioned above indicating advantages and limitations found for FDI in the DAMADICS benchmak.

In this work, we detect, isolate, and classify faults in the DAMADICS benchmark by reconstructing the faults using inverse fuzzy models. While model-based FDI typically relies on observers and residual signals for fault evaluation, the main contribution of this paper is the utilization of inverse models to directly identify the fault responsible for the system behavior without the need for computing residuals. To enhance fault isolation, the faults are grouped into

Table 1: Related works for FDI.

Reference	Methodology	Advantages	Limitations
[30]	Nonlinear observers	Good detection	Time required for residual evolution, not all faults can be isolated.
[9]	Fuzzy models	Automation of fuzzy rules	Fault detection and isolation, to use uncertain information.
[32]	Clustering	Not to require prior information	Fault diagnosis free of model.
[8]	Fuzzy classifier	To isolate the fault magnitude	Computational effort to develop the classifier.
[33]	Artificial neural networks	To improve the fault detection rate or to accelerate the detection time	To collect data for the ANN model, more design to isolate incipient faults.
[35]	Neuro-fuzzy systems	Good detection	Cannot be applied for unknown working conditions, unknown faults cannot be isolated.
[12]	Neuro-fuzzy systems	Detection independent to the fault magnitude	To know the noise distribution
[36]	ANN and decision trees	Fast isolation using past information	To collect data for the ANN model
[37]	Radial-basis functions and fuzzy logic	Temporally informative features with little computational cost.	Average results.
[38]	Self-organizing maps	Classification even with overlapped faults	Some undetectable faults, Time consuming.
[39]	Self-organizing maps	To isolate and identify faults without a data preprocessing	Time consuming.
[42]	Timed automata	Not to require prior information	Only three faults were diagnosed.
[43]	Typicality and eccentricity data analytics	Precision average = 83.30%	Problems to detect incipient faults.
[28]	Hidden Markov model	Early detection	Only three faults were detected.

classes based on their source, and we employed the wavelet transform to highlight incipient faults. The remaining sections of this paper are organized as follows: Section 2 describes the DAMADICS system used as the benchmark for FDI. Section 3 presents the methodology used, including the obtention of fuzzy models (subsection 3.1) and the explanation of how to invert the fuzzy models (subsection 3.2). The results are presented in Section 4, with fault detection results shown in subsection 4.1 and fault isolation presented in subsection 4.2. A comparison with similar applications is also included in the discussion Section 5. Finally, conclusions are provided in Section 6.

2 Benchmark description

The DAMADICS benchmark model, developed by the Research Training Network of the European Commission during the 2000-2004 period under the Framework 5 Human Potential Programme, provides a simulation of the flow dynamics through a valve. The flow rate is determined by the position of a rod, as the rod's movement regulates the flow area inside the valve [20]. This valve, known as an electro-pneumatic actuator, plays a crucial role in controlling the flow through the pipeline installation. It consists of a plate connected to the chamber walls through a flexible diaphragm. The position of the rod is determined by a control signal, which, depending on its value, either connects the chamber with the pneumatic circuit or with the atmosphere. To achieve linear motion of the servomotor stem, a compressible fluid-powered device operates on the flexible diaphragm [20]. To correct any mispositions caused by disturbances, a positioner is employed in the system.

The DAMADICS model offers the capability to simulate 19 different faults, which are listed in Table 2 of [8]. The model itself can be downloaded from `http://diag.mchtr.pw.edu.pl/damadics/`. Notably, the model allows for the simulation of incipient faults, which can often be mistaken for changes in the operating point. These incipient faults can also be utilized to predict the lifespan of industrial actuators [45]. Figure 11 of [20] presents a fault scheme specific to the DAMADICS actuator, illustrating the origin of the faults and the variables involved in identifying each fault. The faults can manifest as either stationary events, such as abrupt faults, or non-stationary signals like random or incipient variations in the values of the supply pressure (fault F_{16}) [46].

The faults in the DAMADICS benchmark can be simulated with various magnitudes, although some of them can only be simulated in an abrupt form. It is important to note that certain cases in Table 3 do not have a physical interpretation and, as a result, were not considered for simulation. These excluded cases may involve faults that are not practically feasible or do not align with real-world scenarios. It is essential to focus on the faults that have meaningful interpretations and are relevant to the system under consideration.

In the study shown in [8], a table was presented that pro-

Table 2: Faults in the DAMADICS model.

Fault origin	Fault	Description	Simulation values
Control valve	F_1	Valve clogging	[0, 1]
	F_2	Valve sedimentation	[0, 1]
	F_3	Valve erosion	[0, 1]
	F_4	Brushing friction	[-1, 1]
	F_5	External leakage	[0, 1]
	F_6	Internal leakage	[0, 1]
	F_7	Critical flow	[0, 1]
Servomotor	F_8	Twisted servomotor's rod	[0, 1]
	F_9	Terminals tightness	[0, 1]
	F_{10}	Servomotor's diaphragm perforation	[0, 1]
	F_{11}	Servomotor's spring fault	[0, 1]
Positioner	F_{12}	Electro-pneumatic transducer fault	[-1, 1]
	F_{13}	Rod displacement sensor fault	[-1, 1]
	F_{14}	Pressure sensor fault	[-1, 1]
	F_{15}	Positioner spring fault	[0, 1]
External faults	F_{16}	Positioner supply pressure fault	[0, 1]
	F_{17}	Unexpected pressure change in valve	[-1, 1]
	F_{18}	Fully or partly opened bypass valves	[0, 1]
	F_{19}	Flow rate sensor fault	[-1, 1]

vides information about the magnitudes of faults that can be simulated in the DAMADICS model. The magnitudes are categorized as small for values ranging from 5% to 35%, medium for values between 35% and 70%, and big for values between 70% and 100% in the case of abrupt faults. Incipient faults, on the other hand, gradually increase their magnitude over time. Detecting incipient faults at an early stage can be challenging, especially when the measured signals are noisy [47].

Table 3: Faults without physical interpretation.

Fault	Small	Medium	Big	Incipient
F_1				☒
F_3		☒	☒	
F_4	☒	☒	☒	
F_8				☒
F_{11}	☒	☒		

The DAMADICS actuator is depicted in Figure 1, where the three main components are distinguished by different colors. The diagram showcases the input and output variables associated with the system. The yellow-highlighted component represents the spring-and-diaphragm pneumatic servomotor, which facilitates the interaction between the control signal and the final movement. To better understand the nomenclature used in Figure 1, please refer to Table 4.

The DAMADICS model was implemented using Simulink®, a popular simulation software. To ensure accurate simulation results, certain considerations were taken into account. The ordinary differential equations (ODEs) describing the system dynamics were solved using the fourth-order Runge-Kutta method (ode4) with a fixed step size of 0.0025 seconds. This choice of solver and step size helps ensure numerical stability and accuracy in the simulation.

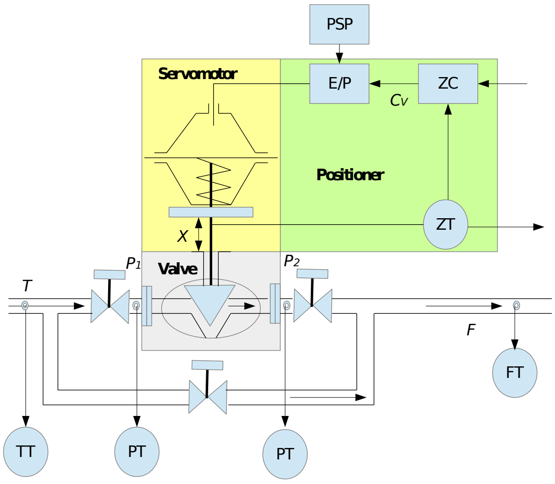


Figure 1: Scheme of DAMADICS actuator.

In addition, it is important to note that the output of the simulated process is subject to noise, which reflects the realistic conditions of industrial systems. All inputs and outputs in the model have band-limited white noise superimposed with a 50 Hz sine wave. The sine bias is set to 2.5% of disturbed signal nominal range and its amplitude is 2.5% of signal nominal range [20]. The presence of noise in the output signals adds a level of complexity to the fault detection and isolation process, as it introduces uncertainty and can affect the accuracy of the results.

Moreover, some specific faults in the DAMADICS model are characterized by a parameter t_{fd}, which repre-

Table 4: Faults in the DAMADICS model.

Nomenclature	Description
E/P	Electro-pneumatic transducer
FT	Flow rate transmitter
PSP	Positioner supply pressure unit
PT	Pressure transmitter
TT	Temperature transmitter
ZC	Internal position controller
ZT	Stem position transmitter

sents the fault development time. This parameter is particularly relevant for simulating incipient faults, which gradually increase in magnitude over time. By incorporating the fault development time parameter, the model can accurately capture the behavior of such incipient faults and their impact on the system dynamics.

Lastly, it is worth mentioning that the sampling time for collecting data from the simulated system is set to 1 second. This choice of sampling time aligns with the sampling rate typically used in the supervisory control and data acquisition (SCADA) systems deployed in real-world industrial environments, such as the sugar factory in Lublin.

These considerations ensure that the DAMADICS model accurately represents the behavior of the physical system and provides a realistic foundation for conducting fault detection and isolation experiments.

3 Methodology used for fault detection and isolation

In the field of fault detection and isolation (FDI), [48] employed an inverse fuzzy model for fault reconstruction. Given the Tennessee Eastman process, which offers numerous state variables for measurement [14, 49], the methodology adopted least angle regression for variable selection [50]. The faults were encoded in terms of amplitude and effectively isolated. However, it should be noted that in certain cases, the FDI process required a minimum of 1000 sample times. Additionally, other FDI techniques have been explored, such as partial least squares [51], elastic net [52], and least absolute shrinkage and selection operator algorithm [53].

3.1 Fuzzy modeling

In order to establish a fuzzy model, it is essential to define the discourse universe of the variables, which determines the fuzzy partition [54]. The ranges of the variables were presented by [9] in its Table 1, and the specific values utilized in this study are summarized in Table 5.

The inversion of the fuzzy model requires a monotonic model [31]. In a study by [55], a fuzzy model of a greenhouse was developed using two membership functions to ensure a monotonic model. The control is achieved by inverting the fuzzy model. In this context, two membership functions are constructed for the fuzzy partition of variables C_V, P_1, P_2, and f, while variable T remains constant throughout the simulation. As a result, the fuzzy model aims to approximate variables X and F.

Table 5: Ranges and units of the variables.

Variable	Range
C_V	0–100%
P_1	0–4000 KPa
P_2	0–4000 KPa
T	0–200°C
X	0–100%
F	0–40 m^3/h

The fuzzy rules can be denoted as [56]:

$$\begin{aligned} \text{If:} \quad & C_V(k) \text{ is } \tilde{A} \text{ and } P_1(k) \text{ is } \tilde{B} \text{ and } P_2(k) \\ & \text{is } \tilde{C} \text{ and } f(k) \text{ is } \tilde{D} \text{ and } X(k) \text{ is } \tilde{E} \ldots \\ & \text{and } F(k) \text{ is } \tilde{F}, \qquad (1) \\ \text{then:} \quad & X_{apr}(k+1) = \theta, \end{aligned}$$

where k is the sample, $\tilde{A}$ is a fuzzy set, and the rule consequent is θ.

The fuzzy value obtained for each variable using a fuzzy set is denoted for example as $\mu_{\tilde{A}_1}(C_V(k))$.

The rule evaluation is:

$$\begin{aligned} \lambda_1(k) &= \mu_{\tilde{A}_1}(C_V(k))\mu_{\tilde{B}_1}(P_1(k))\mu_{\tilde{C}_1}(P_2(k)) \\ &\quad \mu_{\tilde{D}_1}(f(k))\mu_{\tilde{E}_1}(X(k))\mu_{\tilde{F}_1}(F(k)), \qquad (2) \end{aligned}$$

$$\begin{aligned} \lambda_2(k) &= \mu_{\tilde{A}_1}(C_V(k))\mu_{\tilde{B}_1}(P_1(k))\mu_{\tilde{C}_1}(P_2(k)) \\ &\quad \mu_{\tilde{D}_1}(f(k))\mu_{\tilde{E}_1}(X(k))\mu_{\tilde{F}_2}(F(k)), \end{aligned}$$

and so on until to compute $\lambda_{64}(k)$.

The fuzzy partition for each variable is created by employing two membership functions. These functions utilize two limits to achieve a membership value of one. A line connects the maximum value to the zero value, which is positioned below the other limit. This construction ensures that the sum of membership values is always equals to one, thereby guaranteeing the monotonic nature of the fuzzy model [55].

There are a total of 64 fuzzy rules, because each one of the six variables is partitioned into two fuzzy sets. The membership values are arranged in a specific order to invert the fuzzy model. For this reason, the membership values assigned to the rules using $\mu_{\tilde{D}_1}(f(k))$ and $\mu_{\tilde{D}_2}(f(k))$ for the fuzzy partition of the fault $f(k)$ are placed in the last two columns of β. This arrangement results in a matrix $\beta \in \mathbb{R}^{n\times 64}$, which is utilized for the inversion of the fuzzy model. Here, n represents the number of samples used for constructing the fuzzy model.

A normalization factor, denoted as $\Gamma_{i,j}$, can be computed when the summation of membership values does not equal one. It is calculated as $\Gamma_{i,j} = \beta_{i,j}/\sum_{j=1}^{2}\beta(i,j)$. The least squares method is employed to compute the rule consequents θ using the normalized values in Γ. It is worth noting that the temperature variable T was not included in the fuzzy model as it does not appear to be sensitive to faults.

Some fault types can be simulated in each case, and only the ones with a physical interpretation were used for modeling. Finally, θ is computed by using:

$$\theta_F = (\Gamma^T\Gamma)\Gamma^T F_+, \tag{3}$$

where F_+ represents the measure taken with an advance of one sample, i.e. the model must obtain the system behavior for the next sample. Thus, the fuzzy model obtains $F_{fuzzy} = \Gamma\theta_F$. A fuzzy model can be obtained in the same way to approach $X(k+1)$. In that case, the consequents of the rules are in θ_X

Since the fuzzy model estimates two output variables, two sets of consequents are obtained: θ_F for approximating the flow rate F, and θ_X for the displacement of the valve head.

In some cases, the model obtained using θ_F closely approximates the actual variable F. In order to reconstruct a fault, it is necessary for the fault signal to cause a change in the output variables F or X. Moreover, if this change is proportional to the magnitude of the fault, the fault signal can be reconstructed. For instance, in Fig. 2, the fuzzy model obtained to approximate F when fault F_1 occurs at different magnitudes over time is shown. The corresponding fault signal is also depicted, illustrating a noticeable modification in the flow rate behavior. Similarly, in Fig. 3, the same situation is demonstrated. In this case, the fault F_{17} is simulated, including the incipient case. It can be observed that the change in F exhibits similar behavior for different magnitudes.

As mentioned above, a sampling time of one second is used, and the process inputs are normalized sinusoidal signals. This choice allows for modifications in the operating conditions and the inclusion of nonstationary signals. By using signals with different frequencies and amplitudes, the model can capture various dynamics and better represent real-world scenarios.

The inputs are defined as:

$$\begin{aligned} C_V(k) &= 0.5 + 0.25\sin(0.01k), &(4)\\ P_1(k) &= 3500,000 + 175,000\sin(0.1k), &(5)\\ P_2(k) &= 2600,000 + 26,000\sin(k/\pi). &(6)\end{aligned}$$

The temperature is assumed constant $T = 43°C$ [57]. All the variables are noisy, and the pressure measures are the disturbed values obtained from the model.

3.2 Fuzzy model inversion

To invert the fuzzy model, the process variables are evaluated independently of the fault signal $f(k)$. This results in obtaining 32 rules for the inversion process. The consequents, represented by θ_F in this case, are organized into two columns. The values in one column correspond to the consequents obtained using the first limit $\tilde{D}_1$ for f, while the values in the other column correspond to the consequents obtained using the second limit $\tilde{D}_2$, as shown in Equation (1). These two columns are combined to form the matrix $\Theta \in \mathbb{R}^{32\times 2}$, which contains the consequent parameters for the inversion process.

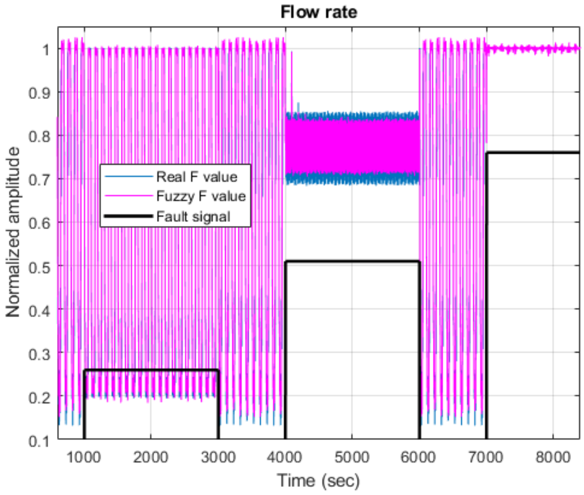


Figure 2: Comparison between flow rate F and its fuzzy approximation under fault F_1.

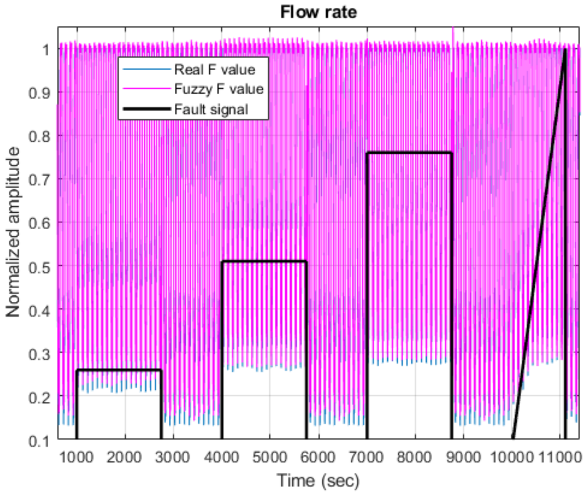


Figure 3: Comparison between flow rate F and its fuzzy approximation under fault F_{17}.

The first step in the fault reconstruction process is to calculate the first center of the fuzzy partition. This is done by multiplying the 32 membership values $\mu(k)$ with the values in the first column of Θ using the Hadamard product. Similarly, the second limit $\tilde{F}_2^*$ is obtained by multiplying the membership values with the values in the second column of Θ, following a similar procedure as Equation (1) shows.

To obtain the membership values for the original signal $F(k)$ using the new fuzzy sets, the signal is evaluated by the new fuzzy sets. These new membership values are then multiplied with the limits used to fuzzify f, which are

$[\tilde{D}_1 \ \ \tilde{D}2]^T$. This process yields two values, which are used to approximate the reconstructed fault signal $fapr$.

Similarly, if we want to invert the fuzzy model used to approach X, the new limits $[\tilde{E}_1^* \ \ \tilde{E}_2^*]$ to fuzzify X are obtained, and the fault reconstruction is:

$$f_{apr}(k) = [\mu_{\tilde{E}_1^*}(X(k)) \ \ \mu_{\tilde{E}_2^*}(X(k))] \ \ [\tilde{D}_1 \ \ \tilde{D}_2]^T. \qquad (7)$$

The fuzzy model inversion can be briefly explained following the next steps:

1. The consequents θ of the 64 fuzzy rules are ordered to form Θ that has as first column the 32 membership values corresponding for the case when the rules evaluated use the first fuzzy set for the partition of fault $f(k)$, i.e. $\tilde{D}_1$. The second column correspond to the 32 rules that contains the membership values when fuzzy set $\tilde{D}_2$ was used.

2. The current variables values are evaluated by their respective fuzzy sets, i.e. we obtain $\mu_{\tilde{A}_1}(C_v(k))$, $\mu_{\tilde{A}_2}(C_v(k))$, $\mu_{\tilde{B}_1}(P_1(k))$, ..., $\mu_{\tilde{F}_2}(F(k))$.

3. Now all this membership valued are use to evaluate a fuzzy rule, the product was used in this step, i.e.

$$\begin{aligned} R_1(k) &= \mu_{\tilde{A}_1}(C_v(k))\mu_{\tilde{B}_1}(P_1(k))\mu_{\tilde{C}_1}(P_2(k)) \\ &\quad \cdots \mu_{\tilde{F}_1}(F(k)) \\ R_2(k) &= \mu_{\tilde{A}_1}(C_v(k))\mu_{\tilde{B}_1}(P_1(k))\mu_{\tilde{C}_1}(P_2(k)) \\ &\quad \cdots \mu_{\tilde{F}_2}(F(k)) \end{aligned}$$

   until,

$$\begin{aligned} R_{32}(k) &= \mu_{\tilde{A}_2}(C_v(k))\mu_{\tilde{B}_2}(P_1(k))\mu_{\tilde{C}_2}(P_2(k)) \\ &\quad \cdots \mu_{\tilde{F}_2}(F(k)) \end{aligned}$$

4. The new centers $\tilde{E}_1^*$ and $\tilde{E}_2^*$ if the fuzzy model for X is inverted, or $\tilde{F}_1^*$ and $\tilde{F}_2^*$, are computed multiplying the evaluation rules of step 3 and the matrix Θ.

5. The model inversion use the fuzzy model of X or F, to obtain the fault f. Then, the signal used, e.g. F is evaluated by the new fuzzy sets with centers $\tilde{F}_1^*$ and $\tilde{F}_2^*$.

6. Finally, the fault approach is obtained by multiplying these last membership values and the centers used initially for the fuzzy partition of fault signal $\tilde{D}_1$ and $\tilde{D}_2$.

To illustrate this procedure Fig. 4 shows how to obtain the new centers used to obtain the fault than generates the current signals behavior.

The fault reconstruction, as depicted in Fig. 5, appears to be noisy and may not immediately reveal the presence of the fault signal. However, we applied a filtering technique to the reconstructed signal, and the fault signal becomes more discernible. In this case, a second-order low-pass filter with two poles at -0.01 is proposed. This filter has a lower frequency of 0.01 and provides an attenuation of $-40dB$ per decade.

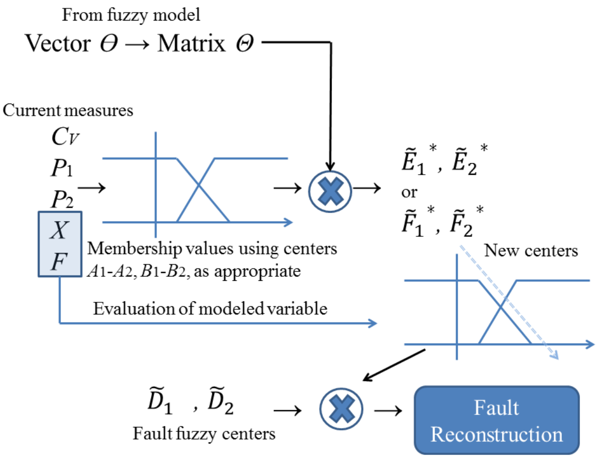

From fuzzy model
Vector $\Theta \rightarrow$ Matrix Θ
Current measures
C_V
P_1
P_2
X
F
Membership values using centers $A1$-$A2$, $B1$-$B2$, as appropriate
Evaluation of modeled variable
$\tilde{E}_1^*$, $\tilde{E}_2^*$
or
$\tilde{F}_1^*$, $\tilde{F}_2^*$
New centers
$\tilde{D}_1$, $\tilde{D}_2$
Fault fuzzy centers
Fault Reconstruction

Figure 4: Fuzzy model inversion.

To implement the filter in a discrete system, a zero-order hold discretization method is utilized [58]. The resulting filter in the Z transform domain is given by the following equation:

$$\frac{f_{filt}(z)}{f_{apr}(z)} = \frac{4.967e-8z+4.934e-5}{z^2-1.98z+0.9802}.$$

due to filtering, a delay is obtained. However, the fault takes some samples to be detected, also when a fault disappears (an intermittent fault), the systems takes some seconds to recover its normal performance.

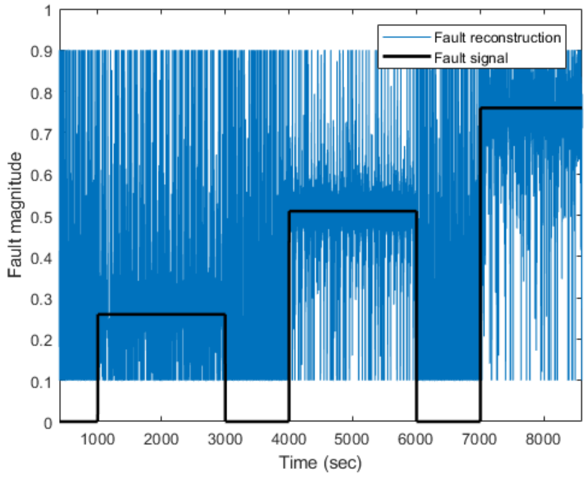


Figure 5: F_{apr} by inverting the fuzzy model of F.

The filtered fault reconstructions obtained by inverting the fuzzy models for fault F_6 and fault F_{17} are illustrated in Fig. 6 and Fig. 7, respectively. In Fig. 6, specific thresholds are depicted, which can be utilized for the detection and classification of fault F_6.

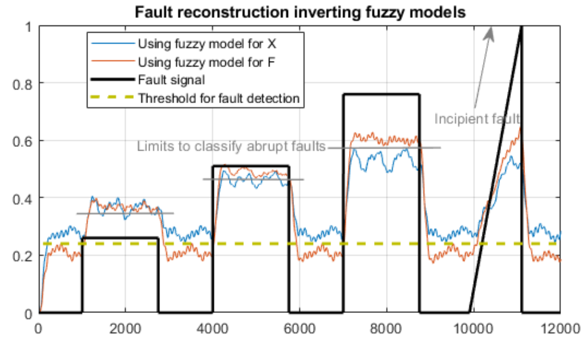


Figure 6: F_6 reconstruction by inverting both fuzzy models.

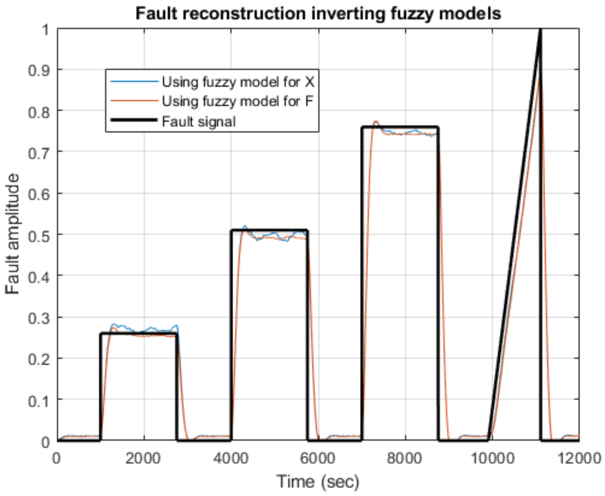


Figure 7: F_{17} reconstruction by inverting both fuzzy models.

In the majority of cases, the fuzzy model for the flow rate F yields the most accurate and reliable fault reconstruction. However, when attempting to reconstruct faults F_8 and F_{15} specifically, the model for the displacement X provides superior results. It is worth noting that if a fault does not manifest itself clearly in the process outputs, it becomes challenging to accurately reconstruct such faults.

Due to the presence of noise in the signals, a threshold-based approach can be employed to detect faults. This involves using the average value of the fault-free signal f_{filt}, along with twice the standard deviation within that specific period, as a reference threshold. It is important to note that the system being analyzed is nonlinear, meaning that the response to a fault may not be directly proportional to the fault magnitude. Consequently, the reconstructed fault values may exhibit different magnitudes compared to the actual faults.

For fault classification purposes, different thresholds can be utilized. In the case of incipient faults, their behavior typically intensifies over time. As such, an alternative approach involves assessing whether the current fault reconstruction demonstrates continuous changes with the same sign. However, due to the presence of noise, the signal f_{filt} tends to fluctuate. Therefore, it is necessary to analyze at least four samples to ensure that the signal consistently exhibits either an increasing or decreasing trend.

Fault isolation is achieved by comparing the responses of different fuzzy models to the current signals obtained. Since each fuzzy model was built for a specific case, the model that yields the highest response (membership value) is used for fault isolation.

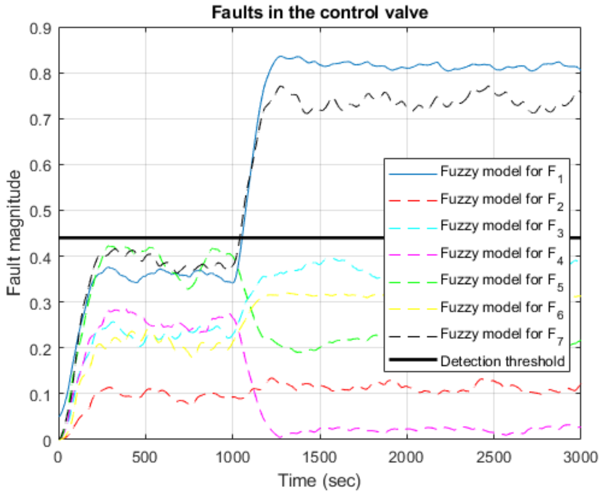


Figure 8: F_1 evaluated by different fuzzy models.

Figure 8 illustrates the fault reconstruction using the fuzzy models for faults $F_1, F_2, \ldots, F_7$, which are associated with faults in the control valve. It is evident that these faults can be detected. However, for fault isolation, it is necessary to evaluate the response of the inverse fuzzy models. Four fuzzy systems are employed for fault isolation. The first fuzzy system evaluates faults $F_1, F_2, \ldots, F_7$ related to the control valve, while the second fuzzy system considers faults $F_8, F_9, \ldots, F_{11}$ in the servomotor. The third fuzzy system incorporates $F_{12}, F_{13}, \ldots, F_{15}$ to describe faults in the positioner. Lastly, the fourth fuzzy system utilizes $F_{16}, F_{17}, \ldots, F_{19}$ for external faults (refer to Table 2).

The FDI proposal involves constructing a fuzzy model to approximate the output signals of the DAMADICS model. By inverting these fuzzy models, the fault signal can be reconstructed using the currently measured signals. Once fault detection is performed, the reconstructed signals are categorized into four sets, corresponding to each fault source: control valve, servomotor, positioner, and external faults. These grouped signals are then inputted into a fuzzy system for fault isolation. The schematic diagram illustrating this technique is presented in Figure 9.

It is worth mentioning that the use of an unknown input observer for fault reconstruction with H_∞ performance has been explored in [59]. Additionally, [60] proposed an intriguing approach that involves inverting certain filters to detect faults in a nonlinear system. This approach shares

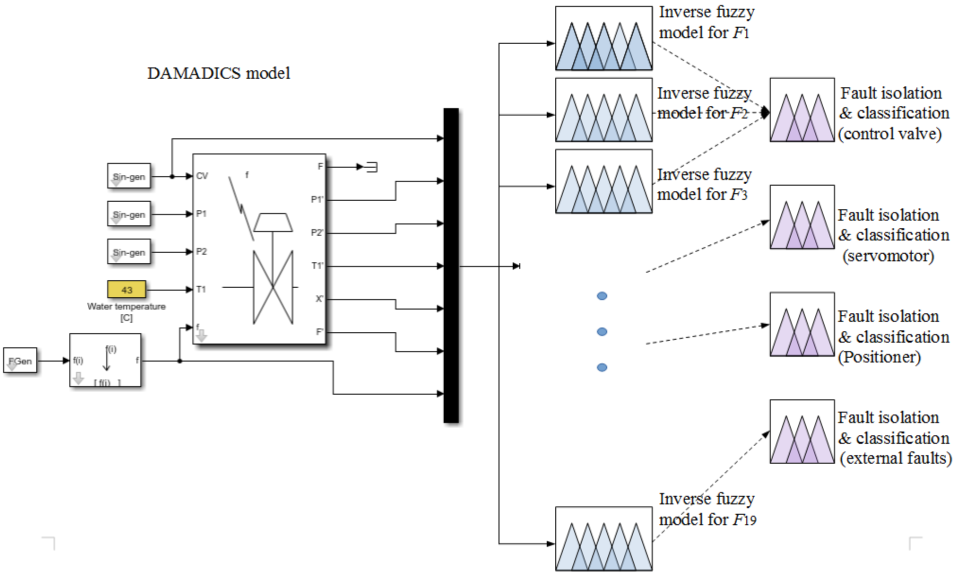

DAMADICS model
Sin-gen
Sin-gen
Sin-gen
43
Water temperature [C]
FGen
CV
P1
P2
T1
F
P1'
P2'
T1'
X'
F'
Inverse fuzzy model for F_1
Inverse fuzzy model for F_2
Inverse fuzzy model for F_3
Inverse fuzzy model for F_{19}
Fault isolation & classification (control valve)
Fault isolation & classification (servomotor)
Fault isolation & classification (Positioner)
Fault isolation & classification (external faults)

Figure 9: Fault detection and isolation scheme.

similarities with the current application, as a fuzzy system can be considered as a type of nonlinear filter.

Similarly, Varkonyi [61] used a genetic algorithm for iterative fuzzy model inversion. His approach involves interchanging one of the inputs with the output in order to invert the fuzzy model. The genetic algorithm was employed to optimize the model inversion process.

By adopting this approach, the fuzzy model can be effectively inverted, enabling the estimation or reconstruction of the input variable based on the observed output. This iterative process using a genetic algorithm allows for the refinement and improvement of the model inversion, leading to more accurate results.

In contrast, [62] employed the system's model to design sliding mode observers. They utilized individual state estimators based on least squares to enable rapid fault reconstruction for FDI applications. This approach differs from the fuzzy model-based technique described earlier.

3.3 Wavelet transform to highlight faults

Wavelets are a powerful tool in signal processing that can effectively detect abrupt changes in signals caused by faults in a system. Unlike stationary events, faults are nonstationary, making wavelets particularly well-suited for fault detection. Wavelet functions are derived from translating (mother wavelet) and scaling (father wavelet) functions to create an orthonormal basis [63]. This allows wavelets to capture both time and frequency information simultaneously, making them ideal for analyzing signals with transient behavior.

The wavelet transform can be utilized to emphasize the existence of a fault within a signal [48]. In the case of discrete wavelet transform, a sufficient number of samples is required for signal decomposition. For instance, if an eighth decomposition level is desired, a minimum of $2^8 = 256$ samples is needed to obtain a meaningful indication of a fault. However, for online fault diagnosis, it is advisable to employ the continuous wavelet transform, where only the selection of the mother wavelet and the scale is necessary [64]. This approach allows for real-time fault detection and characterization without the need for a predefined number of samples or decomposition levels.

The scalogram of the signal X when fault F_1 is depicted in Fig. 10. In order to determine the appropriate scale, the row where all fault occurrences are clearly visible is chosen. Subsequently, the continuous wavelet transform is employed as a filtering technique to localize transients. The wavelet transform plot is shown in Fig. 11. This analysis provides a visual representation of the signal characteristics and highlights the presence of fault-related features.

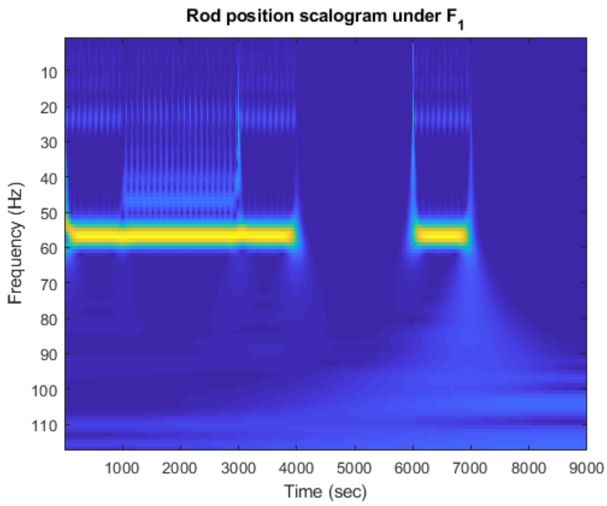


Figure 10: Scalogram to select frequency.

The wavelet transform is particularly effective for detect-

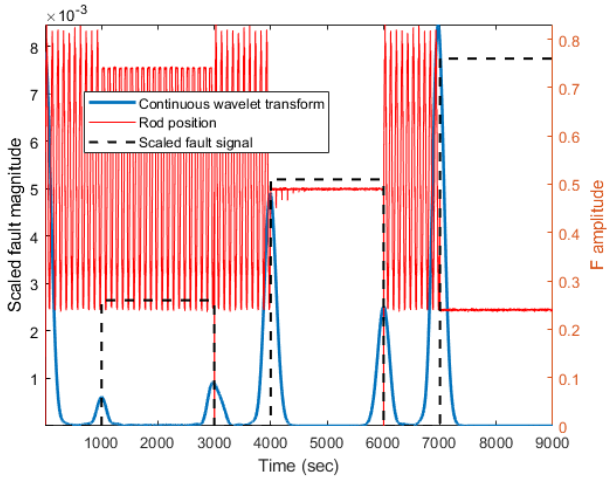


Figure 11: Wavelet transform to localize transients.

ing nonstationary fault signals [65]. It can identify changes in the measurement data that indicate the presence or absence of an event [66]. Therefore, the wavelet transform enables the detection of faults as they occur and as they cease. However, selecting an appropriate scale for fault detection can pose a challenge. It is crucial to consider that the fault may not manifest until at least 200 samples after the simulation or system operation begins, while detection of the fault can occur as early as 50 samples into the process due to the chosen scale of the wavelet.

To streamline the computation of the wavelet transform, a line-based approach is employed instead of shifting the wavelet along the signal being analyzed. This approach involves maintaining a vector that contains the wavelet evaluations, and performing convolutions with the incoming samples. By examining the amplitude of the wavelet transform at the designated scale for each fault, fault classification can be achieved, as demonstrated in Fig. 11. However, fault isolation requires the utilization of the inverse fuzzy model.

The specific wavelet employed for detecting fault levels that are imperceptible to the fuzzy model is the analytic Morlet wavelet. This wavelet is commonly used for edge detection in image processing [67]. The equation representing the wavelet function is as follows:

$$\Psi(t) = e^{-\alpha t^2} e^{2\pi f_c t i}, \qquad (8)$$

where the center frequency is f_c and the width of the Gaussian function is α.

Using (8), the wavelet transform is:

$$W_\Psi f(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \Psi^* \left(\frac{t-b}{a} \right) dt, \qquad (9)$$

where b is the time shift and a is the scale selected regarding the scalogram. Due to the signal being sampled each second, the scale used to obtain Fig. 11 was 63, and then the wavelet function 8 is scaled, shifted, and discretized, then the convolution with the signal X was made to detect transients.

To classify faults such as F_8, F_9, F_{11}, F_{14}, and in some cases F_1, F_{12}, and F_{16}, the wavelet transform is employed to filter the output signals and identify the fault severity. However, it should be noted that wavelets are primarily useful for fault detection [68]. An interesting approach was proposed in [63], where the Daubechies wavelet was utilized to detect faults in the DAMADICS actuator, with a requirement for fault detection within 15 seconds.

4 Results

The simulation incorporates various magnitudes for the fault signals. Magnitudes of 25%, 50%, and 75% were utilized to represent small, medium, and large abrupt faults, respectively. In the case of incipient faults, the magnitude increases gradually from zero to 100% over a duration of 1200 seconds. To classify a fault as incipient, a minimum of four samples is required to assess if the fault exhibits a continuous change over time.

Considering the aforementioned information, it is important to note that the results obtained for fault detection and classification may not be fully representative in certain cases. Furthermore, the detection and isolation of faults often require a specific fault magnitude to be present.

The simulations were conducted by evaluating the downloaded model in MATLAB® R2017b on a laptop equipped with an Intel® Core^{TM} i5-4200U processor running at 2.3 GHz, and with 4 GB of RAM.

4.1 Fault detection

Fault detection refers to the ability to recognize the occurrence of a fault. In this study, an inverse fuzzy model was developed for each individual fault, resulting in a total of 19 fuzzy models. Among these models, the one that provides the best approximation to the behavior of the variable X is selected as the optimal choice. To streamline computational efforts, only one model is utilized for the detection of each specific fault.

Fault detection is accomplished by applying a threshold to the fault reconstruction signal. Due to the presence of nonlinearities, the filtered fault signal (f_{filt}) deviates from zero even in the absence of any simulated fault. Consequently, the threshold is computed by determining the average value of f_{filt} when the fault signal is equal to zero, and then adding double the standard deviation for this condition.

The fault detection rate (FDR) serves as an evaluation metric to assess the effectiveness of the proposed technique. FDR represents the percentage of correctly identified faulty values when f_{filt} exceeds a specified threshold (also known as true positives). It is worth noting that many studies employ datasets containing both faulty and fault-free signals. However, when the fault is no longer present, it may take several samples for the system to return to its normal behavior. This circumstance can lead to an increased number of false alarms. In this work, such cases are also considered and taken into account during the evaluation process.

The evaluation of fault detection performance involves the use of various metrics. These metrics utilize the following terms: true positive (tp), which denotes the correct detection of a sample fault; true negative (tn), which represents the accurate identification of a fault-free signal as normal system behavior; false positive (fp), which refers to the erroneous detection of normal operation data as faulty (false alarm); and false negative (fn), which indicates the misclassification of fault measures as normal system operation. Several characteristics are employed to assess the performance of fault detection and isolation (FDI), including the following metrics [69]:

$$\text{Precision} = \frac{tp}{tp+fp}, \tag{10}$$

$$\text{Memory} = \frac{tp}{tp+fn}, \tag{11}$$

$$\text{Specificity} = \frac{tn}{tn+fp}, \tag{12}$$

$$\text{Accuracy} = \frac{tp+tn}{tp+fp+tn+fn}. \tag{13}$$

Table 6 presents the metrics used to evaluate the fault detection performance of the inverted fuzzy models. When the value in the first column is less than 0.5, it indicates that there are more false alarms than actual fault detections. A memory value below 0.5 implies that the algorithm fails to detect a significant portion of the faulty data. A specificity value lower than 0.5 suggests that the false alarms outnumber the instances of normal behavior without faults. The accuracy parameter is particularly significant as it determines whether real faults were successfully detected, with the algorithm assuming normal operation in the absence of faults.

4.2 Fault isolation

Fault isolation refers to the ability to differentiate between different faults occurring in a process. During the simulation, we observed that certain faults need to reach a certain magnitude to be detected, as their effects may be imperceptible in the measured signals from the DAMADICS model. The fault reconstruction process aids in estimating the magnitude of the fault responsible for the current behavior in the system. Consequently, fault isolation and classification are achieved simultaneously. However, in a different application discussed by [8], a fuzzy classifier was required to classify different fault magnitudes for the same fault type.

In this study, a total of 19 fuzzy models were employed. For fault detection, isolation, and classification, a fuzzy model was specifically developed to approximate the behavior of X when simulating faults F_8 and F_{15} due to the superior results obtained, characterized by fewer false alarms and a higher detection rate. For the remaining faults, we use the fuzzy models focused on approximating the behavior of F.

For fault isolation, it was necessary to establish certain threshold values for fault magnitudes: $F_1 > 31\%$, $F_7 > 41.5\%$, $F_{12} > 42\%$, and $F_{16} > 60\%$. If these conditions were met, fault isolation was found to be 98% accurate in all cases. Additionally, the fault detection characteristics presented in Table 6 exhibited higher values. Simulations were conducted for fault types that have a physical interpretation. Table 7 displays the obtained fault classifications, where fault isolation was determined by comparing the values of f_{filt}, with the larger value indicating the fault present in the system.

The symbol ☒ indicates that the fault was not simulated due to its physical background [57], × denotes that the fault type could not be classified, and ✓ signifies a correct classification. Faults that could not be classified were attributed to f_{filt} exhibiting similar magnitudes regardless of the fault magnitude. Fortunately, the faults were accurately isolated when the fault magnitudes for F_1, F_7, F_{12}, and F_{16} were as previously mentioned.

If the fault magnitude does not reach a certain threshold for detection, it is not considered in the analysis. The fault detection characteristics presented in Table 6 can be summarized as follows in Table 8.

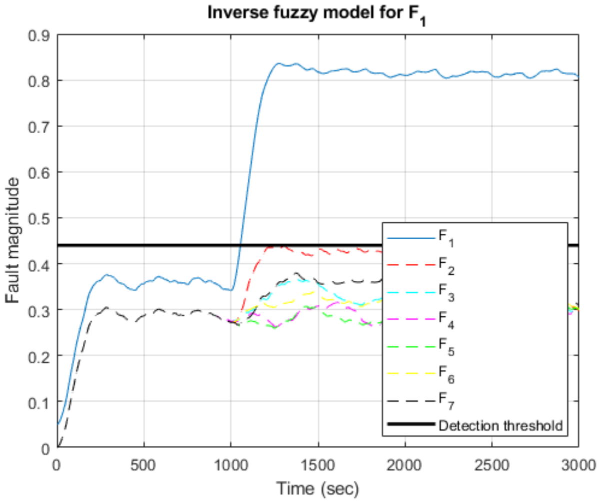


Figure 12: Inverse fuzzy model for F_1 evaluated with different faults.

Similarly to what is shown in Fig. 8 for the simulation and fault reconstruction using different inverse fuzzy models, the fault isolation results are presented in Fig. 12.

To evaluate the performance of fault isolation, a confusion matrix is utilized, as commonly done in fault diagnosis studies [70]. The confusion matrix reflects the accuracy of fault isolation, with higher values along the diagonal indicating successful isolation. In Fig. 13, it can be observed that faults F_8, F_9, F_{13}, F_{14}, and F_{17} pose greater challenges for isolation, despite exhibiting a good detection rate.

Table 6: Fault detection in the DAMADICS model.

Fault origin	Fault	Precision	Memory	Specificity	Accuracy
Control valve	F_1	0.9521	0.6541	0.9344	0.7496
	F_2	0.8824	0.9724	0.8524	0.9163
	F_3	0.8596	0.8610	0.9078	0.8901
	F_4	0.5367	0.6908	0.8950	0.8644
	F_5	0.9258	0.8176	0.9249	0.8669
	F_6	0.9067	0.9759	0.8835	0.9331
	F_7	0.9265	0.6856	0.9359	0.8005
Servomotor	F_8	0.7224	0.1786	0.9094	0.4778
	F_9	0.9086	0.1304	0.9859	0.5215
	F_{10}	0.8116	0.8389	0.7741	0.8089
	F_{11}	0.9159	0.5841	0.9482	0.7693
Positioner	F_{12}	0.9038	0.5994	0.9260	0.7507
	F_{13}	0.9003	0.9818	0.8738	0.9318
	F_{14}	0.2342	0.0312	0.8866	0.4121
	F_{15}	0.8471	0.9778	0.7952	0.8932
External faults	F_{16}	0.7372	0.4718	0.8049	0.6260
	F_{17}	0.8695	0.9964	0.8265	0.9177
	F_{18}	0.9065	0.9659	0.8832	0.9282
	F_{19}	0.8830	0.9908	0.8477	0.9246

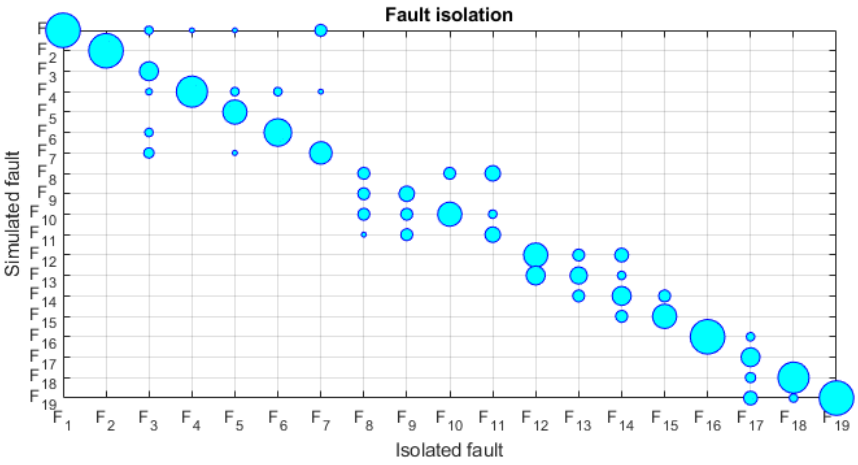


Figure 13: Confusion matrix.

5 Discussion

The obtained fault detection rates are consistent with those reported in the literature. However, the use of inverse fuzzy models contributes significantly to fault isolation, which is not extensively addressed in existing papers focusing on the DAMADICS model. These models provide a means to classify faults based on their magnitude and behavior, enabling comprehensive fault diagnosis. One limitation is the reliance on the model that yields the highest response to ensure accurate fault isolation.

5.1 Comparison of detection rates

The fault detection rate obtained in this study is 84.81%, which significantly outperforms the rates reported in other works, such as 55.45% and 82.37%. This improvement can be attributed to the use of inverse fuzzy models combined with wavelet transform, which enhances the detection of small and abrupt faults that are often missed by conventional methods. Our method demonstrates higher sensitivity and reliability in identifying faults, thereby reducing the risk of undetected issues in chemical processes. A comparison of the obtained results for fault detection is presented in Table 9, alongside the results from [37] and [43]. The best results are highlighted in bold.

5.2 Analysis of computational efficiency

Our approach also shows advantages in terms of computational efficiency. The inversion of fuzzy models is computationally less intensive compared to other advanced tech-

Table 7: Fault classification in the DAMADICS model.

Fault origin	Fault	Small	Medium	Big	Incipient
Control valve	F_1	$\times$	$\checkmark$	$\checkmark$	$\boxplus$
	F_2	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
	F_3	$\boxplus$	$\boxplus$	$\boxplus$	$\checkmark$
	F_4	$\boxplus$	$\boxplus$	$\boxplus$	$\checkmark$
	F_5	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
	F_6	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
	F_7	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
Servomotor	F_8	$\times$	$\times$	$\times$	$\boxplus$
	F_9	$\times$	$\times$	$\times$	$\times$
	F_{10}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
	F_{11}	$\boxplus$	$\boxplus$	$\times$	$\times$
Positioner	F_{12}	$\times$	$\checkmark$	$\checkmark$	$\checkmark$
	F_{13}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
	F_{14}	$\times$	$\times$	$\times$	$\times$
	F_{15}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
External faults	F_{16}	$\times$	$\times$	$\checkmark$	$\checkmark$
	F_{17}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
	F_{18}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
	F_{19}	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$

Table 8: Fault detection for detectable levels in the DAMADICS model.

Fault	Precision	Memory	Specificity	Accuracy
F_1	0.9521	0.9810	0.9606	0.9696
F_7	0.9249	0.9797	0.9571	0.9650
F_{12}	0.8705	0.8242	0.9341	0.8957
F_{16}	0.8580	0.8532	0.9658	0.9439

niques like artificial neural networks or support vector machines. This efficiency is critical in real-time applications where prompt detection and isolation of faults are necessary to prevent process disruptions. The simplicity and lower computational requirements of our method make it suitable for integration into existing industrial systems without requiring significant hardware upgrades. A disadvantage is to require data to obtain the fuzzy models off-line before to use this proposal.

5.3 Robustness against noisy data and incipient faults

One of the key strengths of our approach is its robustness against noisy data and the detection of incipient faults. By employing wavelet transform, our method effectively filters out noise and enhances the signal corresponding to faults highlighting abnormal events. This robustness is crucial in real-world industrial environments where data is often contaminated with noise. Additionally, the capability to detect incipient faults allows for early intervention, preventing minor issues from escalating into major problems. Our method's superior fault isolation accuracy, at 78.85%, further underscores its effectiveness in distinguishing between different fault types, even in challenging conditions. For this reason a low-pass filter was needed to reconstruct the fault signals. However, the filter adds time to detect and isolate faults.

5.4 Improvement over state of the art techniques

Compared to state-of-the-art (SOTA) techniques, our method offers several improvements. While SOTA techniques like deep learning models provide high accuracy, they often require extensive computational resources and large datasets for training. In contrast, our inverse fuzzy model approach achieves competitive accuracy with lower computational demands and without the need for extensive data preprocessing. This balance of accuracy and efficiency makes our method a practical choice for fault detection in the chemical industry.

Among the faults considered, fault F_9 exhibits the lowest detection rate. To enhance the detection of this particular fault, alternative preprocessing methods can be explored instead of relying solely on the wavelet transform, as indicated by the results presented in Table 9. Notably, a detection rate exceeding 0.9 was achieved in the study by [43]. Grouping the fuzzy models based on the possible fault sources allows for a targeted comparison of models for fault isolation.

An intriguing avenue for future research involves developing fault-tolerant control strategies for the valve actuator under consideration, leveraging the insights gained from the fault diagnosis results.

Table 9: Fault detection rates comparison.

Fault	This proposal	[37]	[43]
F_1	0.7480	0.031	**0.9201**
F_2	0.9545	**0.988**	0.8333
F_3	**0.9592**	Not reported	0.3663
F_4	**0.4451**	Not reported	Not reported
F_5	**0.8275**	Not reported	0.7228
F_6	**0.9761**	Not reported	0.7327
F_7	0.9265	0.988	**1.000**
F_8	0.6952	0.121	**0.9333**
F_9	0.5319	Not reported	**0.9130**
F_{10}	0.8392	0.434	**0.9167**
F_{11}	0.6308	0.409	**0.8974**
F_{12}	0.6768	0.111	**0.9302**
F_{13}	**0.9823**	0.880	0.0090
F_{14}	0.4142	Not reported	**0.8076**
F_{15}	**0.9784**	0.373	0.6863
F_{16}	0.6221	0.094	**0.8352**
F_{17}	0.9968	**0.998**	0.8393
F_{18}	**0.9818**	0.834	0.9365
F_{19}	**0.9916**	0.947	0.9716

In conclusion, our study demonstrates that the combination of inverse fuzzy models and wavelet transform provides a robust, efficient, and accurate method for fault detection, isolation, and classification. This approach not only enhances detection rates and computational efficiency but also offers superior robustness against noisy data and incipient faults, making it a valuable tool for maintaining the reliability and safety of chemical processes.

6 Conclusions

In this study, we demonstrate how inverse fuzzy models can be applied to fault reconstruction, facilitating the detection and isolation of faults. We utilized the DAMADICS benchmark, specifically focusing on the electro-pneumatic valve, as our case study for proposing and implementing fault diagnosis techniques. The effectiveness of the fuzzy models in fault detection was evident through the establishment of appropriate thresholds. However, due to the inherent noise present in the output signals, the inverse fuzzy models do not yield a zero output in the absence of faults. To address this issue, a filtering process was employed for fault reconstruction, albeit with the introduction of some delay.

Among the faults considered, F_8 and F_{14} posed the greatest challenges in terms of detection. False alarms were observed with the inverse fuzzy models when simulating F_{15} and F_{16}. However, for other fault types, the fault detection results were comparable to those reported in similar studies. Notably, improved results were achieved for faults $F_3 - F_6$, F_{13}, F_{18}, and F_{19} compared to the literature. Nevertheless, in terms of fault isolation, the inverse fuzzy models exhibited similar behavior across different fault types, often leading to misinterpretation of certain faults such as F_8, F_{13}, F_{14}, and F_{17} as other fault types.

When simulating faults F_1 or F_7, detection was possible, but classification proved challenging, particularly for small abrupt faults. Additionally, in the case of incipient faults, limitations arose due to the physical constraints, resulting in the simulation of only certain fault types. In these cases, a specific fault magnitude had to be applied for effective fault detection. Specifically, fault magnitudes greater than 31% for F_1, 41% for F_7, 42% for F_{12}, and 60% for F_{15} were required. Notably, faults F_8, F_9, and F_{14} could not be classified as small, medium, large, or incipient.

The use of wavelet transform proved beneficial in detecting small faults for F_1 and F_7. However, it did not contribute significantly to fault isolation. One limitation of the proposed methodology is the requirement of measured data during fault conditions to build fuzzy models for fault reconstruction. As future work, the integration of ANNs is planned to facilitate fault prognosis by identifying fault signatures. Additionally, the application of inverse fuzzy fault models holds promise for the development of fault-tolerant control strategies.

Declaration of competing interest

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.

Acknowledgement

The authors would like to thank the Mexican National Science and Technology Council, this paper was possible under grants of the National Laboratories program, (LANAVEX, CONAHCyT).

References

[1] M. Mansouri, M. Nounou, H. Nounou, and N. Karim, “Kernel PCA-based GLRT for nonlinear fault detection of chemical processes,” *Journal of Loss Prevention in the Process Industries*, vol. 40, pp. 334–347, 2016. https://doi.org/10.1016/j.jlp.2016.01.011

[2] A. Imanaka, T. Murayama, S. Nishikizawa, and A. Nagaoka, “Local governments’ response to accidents in chemical factories in Japan: Focus on petroleum industrial complexes special accident prevention areas,” *International Journal of Disaster Risk Reduction*, vol. 74, p. 102880, 2022. https://doi.org/10.1016/j.ijdrr.2022.102880

[3] C. Chen and G. Reniers, “Chemical industry in China: The current status, safety problems, and pathways for future sustainable development,” *Safety Science*, vol. 128, p. 104741, 2020. https://doi.org/10.1016/j.ssci.2020.104741

[4] J. N. P. Nogueira, P. A. Melo, and M. B. de Souza Jr., “Faulty scenarios in sour water treatment units: Simulation and AI-based diagnosis,” *Process Safety and Environmental Protection*, vol. 165, pp. 716–727, 2022. https://doi.org/10.1016/j.psep.2022.07.043

[5] S. Bansal and J. T. Selvik, “Investigating the implementation of the safety-diagnosability principle to support defence-in-depth in the nuclear industry: A Fukushima Daiichi accident case study,” *Engineering Failure Analysis*, vol. 123, p. 105315, 2021. https://doi.org/10.1016/j.engfailanal.2021.105315

[6] A. S. Yeardley, J. O. Ejeh, L. Allen, S. F. Brown, and J. Cordiner, “Integrating machine learning techniques into optimal maintenance scheduling,” *Computers & Chemical Engineering*, vol. 166, p. 107958, 2022. https://doi.org/10.1016/j.compchemeng.2022.107958

[7] R. Doraiswami and L. Cheded, *Fault Diagnosis and Detection*, ch. Fault detection and isolation, pp. 1–27. IntechOpen, 2017. https://doi.org/10.5772/67870

[8] C. D. Bocaniala and J. S. da Costa, “Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem,” *Control Engineering Practice*, vol. 14, pp. 653–669, 2006. https://doi.org/10.1016/j.conengprac.2005.06.008

[9] N. Hadroug, A. Hafaifa, and A. Daoudi, “Valve actuator fault classification based on fuzzy system using the DAMADICS model,” in *International Conference on Applied Automation and Industrial Diagnostics* (U. of Djelfa, ed.), (Algeria), p. 0185, March 2015.

[10] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, “A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process,” *Journal of Process Control*, vol. 22, pp. 1567–1581, 2012. https://doi.org/10.1016/j.jprocont.2012.06.009

[11] A. Kowsalya and B. Kannapiran, “Principal component analysis based approach for fault diagnosis in pneumatic valve using DAMADICS benchmark simulator,” *International Journal of Research in Engineering and Technology*, vol. 3(7), pp. 702–707, 2014. https://doi.org/10.15623/IJRET.2014.0319125

[12] F. J. Uppal, R. J. Pattona, and M. Witczak, “A neuro-fuzzy multiple-model observer approach to robust fault diagnosis based on the DAMADICS benchmark problem,” *Control Engineering Practice*, vol. 14, 2006. https://doi.org/10.1016/j.conengprac.2005.04.015

[13] J. Ding, R. Alroobaea, A. M. Baqasah, A. Althobaiti, R. Miglani, H. S. Gill, “Big bata intelligent collection and network failure analysis based on artificial intelligence,” *Informatica*, vol. 46, pp. 383-392, 2022. https://doi.org/10.31449/inf.v46i3.3866

[14] S. Xiong, L. Zhou, Y. Dai, and X. Ji, “Attention-based long short-term memory fully convolutional network for chemical process fault diagnosis,” *Chinese Journal of Chemical Engineering*, vol. 56, pp. 1-14, 2022.https://doi.org/10.1016/j.cjche.2022.06.029

[15] R. Qin and J. Zhao, “Adaptive multiscale convolutional neural network model for chemical process fault diagnosis,” *Chinese Journal of Chemical Engineering*, vol. 50, pp. 398–411, 2022. https://doi.org/10.1016/j.cjche.2022.10.001

[16] G. C. Silva, E. E. O. Carvalho, and W. M. Caminhas, “An artificial immune systems approach to case-based reasoning applied to fault detection and diagnosis,” *Expert Systems with Applications*, vol. 140, p. 112906, 2020. https://doi.org/10.1016/j.eswa.2019.112906

[17] Z. Xie, J. Chen, Y. Feng, and S. He, “Semi-supervised multi-scale attention-aware graph convolution network for intelligent fault diagnosis of machine under extremely-limited labeled samples,” *Journal of Manufacturing Systems*, vol. 64, pp. 561–

577, 2022. https://doi.org/10.1016/j.jmsy.2022.08.007

[18] G. Hong and D. Suh, "Mel spectrogram-based advanced deep temporal clustering model with unsupervised data for fault diagnosis," *Expert Systems With Applications*, vol. 217, p. 119551, 2023. https://doi.org/10.1016/j.eswa.2023.119551

[19] Y. Xu, X. Zeng, S. Bernard, and Z. He, "Data-driven prediction of neutralizer pH and valve position towards precise control of chemical dosage in a wastewater treatment plant," *Journal of Cleaner Production*, vol. 348, p. 131360, 2022. https://doi.org/10.1016/j.jclepro.2022.131360

[20] M. Bartyś, R. Patton, M. Syfert, S. de las Heras, and J. Quevedo, "Introduction to the DAMADICS actuator FDI benchmark study," *Control Engineering Practice*, vol. 14(6), no. 6, pp. 577–596, 2006. https://doi.org/10.1016/j.conengprac.2005.06.015

[21] P. Subbaraj and B. Kannapiran, "Fault detection and diagnosis of pneumatic valve using adaptive neuro-fuzzy inference system approach," *Applied Soft Computing*, vol. 19, pp. 362–371, 2014. https://doi.org/10.1016/j.asoc.2014.02.008

[22] D. Saravanakumar, B. Mohan, and T. Muthuramalingam, "A review on recent research trends in servo pneumatic positioning systems," *Precision Engineering*, vol. 49, pp. 481–492, 2017. https://doi.org/10.1016/j.precisioneng.2017.01.014

[23] S.-H. Wang, T.-W. Ni, and Z.-G. Yang, "Failure analysis on abnormal blockage of electro-hydraulic servo valve in digital electric hydraulic control system of 125 MW thermal power plant," *Engineering Failure Analysis*, vol. 123, p. 105294, 2021. https://doi.org/10.1016/j.engfailanal.2021.105294

[24] K. Zhang, J. Yao, and T. Jiang, "Degradation assessment and life prediction of electro-hydraulic servo valve under erosion wear," *Engineering Failure Analysis*, vol. 36, pp. 284–300, 2014. https://doi.org/10.1016/j.engfailanal.2013.10.017

[25] J. Shi, J. Yi, Y. Ren, Y. Li, Q. Zhong, H. Tang, and L. Chen, "Fault diagnosis in a hydraulic directional valve using a two-stage multi-sensor information fusion," *Measurement*, vol. 179, p. 109460, 2021. https://doi.org/10.1016/j.measurement.2021.109460

[26] F. Khan, M. T. Amin, V. Cozzani, and G. Reniers, *Methods in Chemical Process Safety*, ch. Domino effect: Its prediction and prevention - An overview, pp. 1–35. Elsevier, 2021. https://doi.org/10.1016/bs.mcps.2021.05.001

[27] U. Pal, G. Mukhopadhyay, and S. Bhattacharya, "Failure analysis of spring of hydraulic operated valve," *Engineering Failure Analysis*, vol. 95, pp. 191–198, 2019. https://doi.org/10.1016/j.engfailanal.2018.09.013

[28] G. M. de Almeida and S. W. Park, "Fault detection and diagnosis in the DAMADICS benchmark actuator system - a hidden Markov model approach," *IFAC Proceedings Volumes*, vol. 41(2), no. 2, pp. 12419–12424, 2008. 17th IFAC World Congress. https://doi.org/10.3182/20080706-5-KR-1001.02102

[29] R. B. di Capaci and C. Scali, "Review and comparison of techniques of analysis of valve stiction: From modeling to smart diagnosis," *Chemical Engineering Research and Design*, vol. 130, pp. 230–265, 2018. https://doi.org/10.1016/j.cherd.2017.12.038

[30] V. Puig, A. Stancu, and J. Quevedo, "Robust fault isolation using non-linear interval observers: The DAMADICS benchmark case study," in *16th Triennial World Congress* (Elsevier, ed.), (Prague, Czech Republic), pp. 293–298, 2005. https://doi.org/10.3182/20050703-6-CZ-1902.01851

[31] R. Babuška, *Fuzzy Modeling for Control*. Aachen, Germany: International Series in Intelligent Technologies, 1998. https://doi.org/10.1007/978-94-011-4868-9

[32] A. Lemos, W. Caminhas, and F. Gomide, "Adaptive fault detection and diagnosis using an evolving fuzzy classifier," *Information Science*, vol. 220, pp. 64–85, 2013. https://doi.org/10.1016/j.ins.2011.08.030

[33] Y. Kourd, D. Lefebvre, and N. Guersi, "FDI with neural network models of faulty behaviours and fault probability evaluation: Application to DAMADICS," in *Supervision and Safety of Technical Processes* (I. F. of Automatic Control, ed.), (Mexico City, Mexico), pp. 744–749, August 2012. https://doi.org/10.3182/20120829-3-MX-2028.00106

[34] J. Sarkar, Z. H. Prottoy, M. T. Bari, and M. A. A. Faruque, "Comparison of ANFIS and ANN modeling for predicting the water absorption behavior of polyurethane treated polyester fabric," *Heliyon*, vol. 7(9), no. 9, p. e08000, 2021. https://doi.org/10.1016/j.heliyon.2021.e08000

[35] V. Gomathi, V. Elakkiya, R. Valarmathi, and K. Ramkumar, "Actuator fault detection using adaptive neuro fuzzy approach for DAMADICS benchmark," *Research Journal of Pharmaceutical, Biological and Chemical Sciences*, vol. 7(1), no. 1, pp. 628–635, 2016.

[36] A. Andrade, K. Lopes, B. Lima, and A. Maitelli, "Development of a methodology using artificial neural network in the detection and diagnosis of faults for pneumatic control valves," *Sensors*, vol. 21, p. 853, 2021. `https://doi.org/10.3390/s21030853`

[37] A. R. C. Oliveira and J. M. G. S. da Costa, "Hierarchic fault diagnosis by pattern - recognition approaches applied to DAMADICS benchmark," in *18th World Congress The International Federation of Automatic Control* (I. F. of Automatic Control, ed.), (Milano, Italy), pp. 7737–7742, August 2011. `https://doi.org/10.3182/20110828-6-IT-1002.03638`

[38] A. Katunin, M. Amarowicz, and P. Chrzanowski, "Faults diagnosis using self-organizing maps: A case study on the DAMADICS benchmark problem," in *Federated Conference on Computer Science and Information Systems*, pp. 1673–1681, 2015. `https://doi.org/10.15439/2015F26`

[39] T. Chopra and J. Vajpai, "Classification of faults in DAMADICS benchmark process control system using self organizing maps," *International Journal of Soft Computing and Engineering*, vol. 1(3), no. 3, pp. 85–90, 2011.

[40] V. Palade, C. D. Bocaniala, and L. Jain, *Computational Intelligence in Fault Diagnosis*. Advanced Information and Knowledge Processing, United States of America: Springer-Verlag London, 2006.

[41] C. D. Bocaniala and V. Palade, *Computational intelligence methodologies in fault diagnosis: Review and state of the art, Computational intelligence in fault diagnosis*, ch. Computational intelligence methodologies in fault diagnosis: Review and state of the art, pp. 1–36. United States of America: Springer-Verlag London, 2006. `https://doi.org/10.1007/978-1-84628-631-5_1`

[42] P. Supavatanakul, J. Lunze, V. Puig, and J. Quevedo, "Diagnosis of timed automata: Theory and application to the DAMADICS actuator benchmark problem," *Control Engineering Practice*, vol. 14, pp. 609–619, 2006. `https://doi.org/10.1016/j.conengprac.2005.03.028`

[43] C. Gomes-Bezerra, B. S. Jales-Costa, L. A. Guedes, and P. P. Angelov, "An evolving approach to unsupervised and real-time fault detection in industrial processes," *Expert Systems With Applications*, vol. 63, pp. 134–144, 2016. `https://doi.org/10.1016/j.eswa.2016.06.035`

[44] X. Li, X. Yang, Y. Yang, I. Bennett, and D. Mba, "A novel diagnostic and prognostic framework for incipient fault detection and remaining service life prediction with application to industrial rotating machines," *Applied Soft Computing*, vol. 82, p. 105564, 2019. `https://doi.org/10.1016/j.asoc.2019.105564`

[45] Y. Langeron, A. Grall, and A. Barros, "Actuator lifetime management in industrial automation," *IFAC Proceedings Volumes*, vol. 45(20), no. 20, pp. 642–647, 2012. `https://doi.org/10.3182/20120829-3-MX-2028.00111`

[46] R. Kosturkov, V. Nachev, and T. Titova, "Diagnosis of pneumatic systems on basis of time series and generalized feature for comparison with standards for normal working condition," *TEM Journal*, vol. 10(1), no. 1, pp. 183–191, 2021. `https://api.semanticscholar.org/CorpusID:233904474`

[47] J. Yang and C. Delpha, "An incipient fault diagnosis methodology using local Mahalanobis distance: Fault isolation and fault severity estimation," *Signal Processing*, vol. 200, p. 108657, 2022. `https://doi.org/10.1016/j.sigpro.2022.108657`

[48] M. A. Márquez-Vera, L. E. Ramos-Velasco, O. López-Ortega, N. S. Zúñiga-Peña, J. C. Ramos-Fernández, and R. M. Ortega-Mendoza, "Inverse fuzzy fault model for fault detection and isolation with least angle regression for variable selection," *Computers & Industrial Engineering*, vol. 159, p. 107499, 2021. `https://doi.org/10.1016/j.cie.2021.107499`

[49] J. Vosloo, K. R. Uren, G. van Schoor, L. Auret, and H. Marais, "Exergy-based fault detection on the Tennessee Eastman process," *IFAC-PapersOnLine*, vol. 53(2), no. 2, pp. 13713–13720, 2020. `https://doi.org/10.1016/j.ifacol.2020.12.875`

[50] L. Zhang and K. Li, "Forward and backward least angle regression for nonlinear system identification," *Automatica*, vol. 53, pp. 94–102, 2015. `https://doi.org/10.1016/j.automatica.2014.12.010`

[51] R. Fazai, M. Mansouri, K. Abodayeh, H. Nounou, and M. Nounou, "Online reduced kernel PLS combined with GLRT for fault detection in chemical systems," *Process Safety and Environmental Protection*, vol. 128, pp. 228–243, 2019. `https://doi.org/10.1016/j.psep.2019.05.018`

[52] F. Cannarile, M. Compare, P. Baraldi, G. Diodati, V. Quaranta, and E. Zio, "Elastic net multinomial logistic regression for fault diagnostics of on-board aeronautical systems," *Aerospace Science and Technology*, vol. 94, p. 105392, 2019. `https://doi.org/10.1016/j.ast.2019.105392`

[53] H. Lee, C. Kim, S. Lim, and J. M. Lee, "Data-driven fault diagnosis for chemical processes using transfer entropy and graphical LASSO," *Computers & Chemical Engineering*, vol. 142, p. 107064, 2020.

https://doi.org/10.1016/j.compchemeng.2020.107064

[54] T. Ross, *Fuzzy Logic with Engineering Applications*. West Sussex: John Wiley & Sons, Ltd., 2008.

[55] M. A. Márquez-Vera, J. C. Ramos-Fernández, L. F. Cerecero-Natale, F. Lafont, J.-F. Balmat, and J. I. Esparza-Villanueva, "Temperature control in a MISO greenhouse by inverting its fuzzy model," *Computers and Electronics in Agriculture*, vol. 124, pp. 168–174, 2016. https://doi.org/10.1016/j.compag.2016.04.005

[56] D. Faouzi, "Optimization, modeling and simulation of microclimate and energy management of the greenhouse by modeling the associated heating and cooling systems and implemented by a fuzzy logic controller using artificial intelligence," *Informatica*, vol. 41, pp. 317-331.

[57] M. Bartys and M. Syfert, *Using Damadics Actuator Benchmark Library*. Warsaw University if Technology, 00-661 Warsaw, Poland, 1.22 ed., April 2002. Simulink Library Help File.

[58] X. M. Zhang and Q. L. Han, "Output feedback stabilization of networked control systems with a logic zero-order-hold," *Information Sciences*, vol. 381, pp. 78–91, 2017. https://doi.org/10.1016/j.ins.2016.11.009

[59] H. Dimassi, "A novel fault reconstruction and estimation approach for a class of systems subject to actuator and sensor faults under relaxed assumptions," *ISA Transactions*, vol. 111, pp. 192–210, 2021. https://doi.org/10.1016/j.isatra.2020.10.063

[60] H. Kazemi and A. Yazdizadeh, "Fault reconstruction in a class of nonlinear systems using inversion-based filter," *Nonlinear Dynamics*, vol. 85, pp. 1805–1814, 2016. https://doi.org/10.1007/s11071-016-2796-z

[61] A. R. Várkonyi-Kóczy, A. Álmos, and T. Kovácsházy, "Genetic algorithms in fuzzy model inversion," in *International Fuzzy Systems Conference Proceedings* (IEEE, ed.), (Seul, Korea), pp. 1421–1426, August 1999. https://doi.org/10.1109/FUZZY.1999.790112

[62] M. Shakarami and K. Esfandiari, "Rapid fault reconstruction using a bank of sliding mode observers," *Journal of the Franklin Institute*, vol. 359(18), pp. 11229-11255, 2022. https://doi.org/10.1016/j.jfranklin.2022.09.012

[63] U. Libal and Z. Hasiewicz, "Wavelet based rule for fault detection," *IFAC PapersOnLine*, vol. 51(24), pp. 255–262, 2018. https://doi.org/10.1016/j.ifacol.2018.09.585

[64] M. Jalayer, C. Orsenigo, and C. Vercellis, "Fault detection and diagnosis for rotating machinery: A model based on convolutional LSTM, fast Fourier and continuous wavelet transforms," *Computers in Industry*, vol. 125, p. 103378, 2021. https://doi.org/10.1016/j.compind.2020.103378

[65] Z. Feng and X. Chen, "Adaptive iterative generalized demodulation for nonstationary complex signal analysis: Principle and application in rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 110, pp. 1–27, 2018. https://doi.org/10.1016/j.ymssp.2018.03.004

[66] V. M. Pukhova, T. V. Kustov, and G. Ferrini, "Time-frequency analysis of non-stationary signals," in *Conference of Russian Young Researchers in Electrical and Electronic Engineering* (IEEE, ed.), (Moscow and St. Petersburg, Russia), pp. 1141–1145, January 2018. 10.1109/IVForum.2017.8246088

[67] C. Li, Y. Huang, and L. Zhu, "Color texture image retrieval based on Gaussian copula models of Gabor wavelets," *Pattern Recognition*, vol. 64, pp. 118–129, 2017. https://doi.org/10.1016/j.patcog.2016.10.030

[68] K. Vernekar, H. Kumar, and K. V. Gangadharan, "Gear fault detection using vibration analysis and continuous wavelet transform," *Procedia Materials Science*, vol. 5, pp. 1846–1852, 2014. https://doi.org/10.1016/j.mspro.2014.07.492

[69] K. Yan and X. Zhou, "Chiller faults detection and diagnosis with sensor network and adaptive 1D CNN," *Digital Communications and Networks*, vol. 8, pp. 531–539, 2022. https://doi.org/10.1016/j.dcan.2022.03.023

[70] W. M. Salilew, Z. A. A. Karim, and T. A. Lemma, "Investigation of fault detection and isolation accuracy of different machine learning techniques with different data processing methods for gas turbine," *Alexandria Engineering Journal*, vol. 61(12), no. 12, pp. 12635–12651, 2022. https://doi.org/10.1016/j.aej.2022.06.026

Evaluation of Manifold Dual Contouring Algorithms Based on *k*-d tree and Octree Data Structures

Thabang Ramaijane, Hlomani Hlomani, Irina Zlotnikova, and Thabiso Maupong
Department of Computing and Informatics, School of Pure and Applied Sciences, Botswana International University of Science and Technology, Botswana
E-mail: thabang.ramaijane@gmail.com, hlomanihb@biust.ac.bw, zlotnikovai@biust.ac.bw, maupongt@biust.ac.bw

Keywords: dual contouring, isosurface extraction, k-d trees, octrees, adaptive data structures, simplification errors thresholds, quadratic error functions

Received: November 14, 2023

This study evaluated the performance and efficiency of manifold dual contouring algorithms using k-d trees and octrees, therefore, addressing a critical gap in comparative analysis of these data structures. Despite the popularity of k-d trees and octrees in isosurface extraction, their performance has not been empirically compared. This research specifically focuses on visualization quality, performance metrics, and efficiency across various simplification error thresholds. A comprehensive comparative analysis was conducted to identify the conditions under which each data structure is most suitable. Both algorithms employed the manifold vertex clustering scheme for dual contouring of isosurfaces. The methodology involved evaluating visual output, build time, extraction time, and efficiency based on triangle counts during the simplification process. Computational experiments demonstrated that the octree-based algorithm is superior for rendering large models, producing an average of 20% more visual detail due to a higher triangle count. In contrast, the k-d tree-based algorithm showed a 40% reduction in build time and a 35% reduction in extraction time, making it more efficient for processing large implicit models by reducing geometric complexity. These findings provide metrics to assist researchers and practitioners in selecting the most suitable adaptive data structure for achieving optimal simplification results in manifold dual contouring algorithm implementations.

Povzetek: Narejena je primerjava algoritmov za 3D obrobe, ki uporabljajo k-d drevesa in okt-drevesa. Zadnji omogoča več podrobnosti, k-d drevo pa hitrejšo obdelavo in gradnjo.

1 Introduction

An *isosurface* is a group of contour lines (*isolines*) in three dimensions where each of these isolines is made of connected points of a fixed scalar value *(an isovalue)* [1]. The purpose of isosurface extraction is to visualize these contour lines in three dimensions (3D) [1]. Applications of 3D visualization include virtual reality [2], [3], augmented reality [4], and computer-aided architectural design [5]. When visualising a 3D dataset, an isosurface extracting algorithm must handle the processing of the geometry of that particular dataset [6]. Isosurface extraction algorithms are implicit surface representation techniques [7].

The underlying surface geometry of an object must be represented as accurately as possible [8]. One of the first algorithms for the representation of high resolution isosurfaces, *the marching cubes (MC)* algorithm by Lorensen and Cline [9], had demonstrated inconsistencies in the way it generated the final visual depictions of extracted isosurfaces. Numerous attempts have been made up to date to overcome the weaknesses of this foundational algorithm. Schaefer and Warren [10] proposed a method for contouring an implicit function using a grid topologically dual to structured grids such as octrees named *dual marching cubes*. By aligning the vertices of the dual grid with the features of the implicit function, Schaefer and Warren were able to reproduce thin features of the extracted surface without excessive subdivision required by the MC or dual contouring (DC) methods. Lee et al. [11] revised the triangulated cubes of the MC algorithm in order to regularize the connectivity of the isosurface mesh and, therefore, to maximize the valence of six vertices. Jin et al. [12] pointed out that the weaknesses of the MC algorithm came from the use of surface configurations of cubes, including wrong surface production and hole generation. In response to that, Jin et al. proposed an improvement of the MC method by reassigning a value of zero to vertices inside the surface and a value of one to vertices on the surface, and then redefining 15 typical configurations considered in the marching cubes method [12]. Strand and Stelldinger [13] modified the MC algorithm by applying it to the face-centered cubic grid meaning that the local configurations considered when extracting the local surface patches were not cubic. Xu et al. [14] advanced the traditional MC algorithm by (1) replacing the cube edge linear interpolation with the midpoint selection, (2) using the index of the intersection point to avoid repeated calculation and (3) contracting edge to reduce the number of triangular patches. Vignoles et al. [15] proposed a simplified MC algorithm in which the surface consisted of triangles composed from vertices of the regular 3D grid on which the processed data was defined. Du et al. [16] made suggestions on enhancement of the original MC algorithm focusing mainly on improving efficiency and topological accuracy of the original algorithm. Greß and Klein [17] used the *k*-d trees for the representation of implicit objects. The *k*-d tree grid was built on the assumption that an appropriate grid partitioning criterion for subdividing splitting planes was determined. Construction of the grid took a top-down routine as follows. For every subdivision step, the active grid edges had to be detected, and their related intersection information was computed. To achieve two-manifold and topologically correct representations, the subdivision scheme maintained a list of faces in the cell. This allowed faces contained in the list to be clipped against the splitting plane throughout each subdivision step. These faces were transmitted to the corresponding sub-cells after being clipped. After generating each vertex during the clipping process, a record was made of the axes corresponding to the planes clipped against them. However, like other previous studies, Greß and Klein [17] did not compare the performance of the *k*-d tree-based algorithm with octree-based algorithms. This paper, therefore, discusses the use of the *k*-d tree as the adaptive data structure alternative to the octree for the representation of isosurfaces extracted from implicit models. Although the octree data structures have already been proposed and implemented for achieving topological improvements in several reviewed studies, the focus of this paper is on evaluating the algorithms' performance in accelerating isosurface extraction under various topology simplification thresholds. The evaluation of both algorithms allowed to determine which data structure would be most adaptive depending on the values of simplification error thresholds. The rest of the paper is organized as follows. Section 2 discusses the fundamental concepts and mathematical equations necessary for the understanding of the material presented in this paper. It also provides the detailed analysis of similar studies. Section 3 discusses the proposed approach. Section 4 outlines the methodology for implementation and evaluation of the algorithms. The results are presented in Section 5. In Section 6, the results are discussed and compared with similar reviewed studies. Finally, the conclusions, limitations and future work are considered in Section 7.

2 Literature review

This section lays a foundation for a better understanding of the material presented in this paper by discussing the fundamental concepts and mathematical equations, as well as providing an overview of existing similar studies.

2.1 Background to the study

This subsection defines the fundamental concepts, such as adaptive data structures and quadratic error functions (QEFs), and discusses their role in isosurface simplification strategies.

2.1.1 Adaptive data structures

The adaptive data structures discussed throughout the paper include *k*-d trees and octrees. These structures simplify isosurface extraction on uniform grids, allowing the simplified grid to adapt to the increasing size of the implicit models being processed. The adaptive nature of these data structures ensures that they can efficiently handle large datasets, making them suitable for applications requiring dynamic and scalable isosurface extraction.

A *k*-d tree is a hierarchical multi-dimensional data structure used for subdividing volume datasets and accelerating isosurface extraction [17]. It subdivides the volume one dimension at a time, resulting in binary partitions at each stage. This method allows the *k*-d tree to be applied in any orientation based on the chosen plane, without a predefined traversal sequence, making it particularly useful for adaptive processing and efficient representation of complex volumetric data. The *k*-d tree is a binary tree where every node is a k-dimensional point [18]. The *k*-d tree performs a binary partition in one dimension at a time. Figure 1 illustrates a typical *k*-d tree volume subdivision.

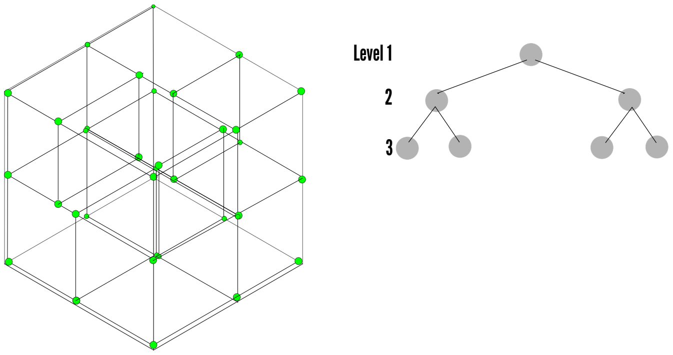


Figure 1: *K*-d tree volume subdivision

An octree is a hierarchical data structure primarily used to represent a decomposed three-dimensional volume, with each node corresponding to a specific sub-volume [19]. Each subdivision produces eight octants, being divided as long as the root node represents the whole volume (Figure 2). An octree simultaneously subdivides a volume in three dimensions, unlike the *k*-d tree which performs a binary partition in one dimension at a time. This data structure is typically employed in isosurface extraction applications to partition uniform and rectilinear grids into subregions, with each leaf node representing a single grid cube. The adaptive nature of the octree allows it to handle large datasets efficiently, as demonstrated by Ju et al.[20].

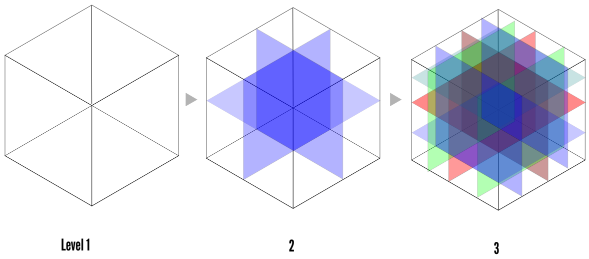


Figure 2: Octree volume subdivision - Levels 1 to 3

2.1.2 Isosurface simplification strategies for adaptive data structures

Simplifying isosurfaces for topologically correct surface representations of volumetric or implicit datasets can be effectively and efficiently achieved using adaptive data structures (i.e., *k*-d trees and octrees). However, the approaches can vary between different implementations.

The three approaches considered in this subsection include (1) error-constrained *k*-d tree grid simplification, (2) octree-based simplification using QEFs, and (3) octree-based simplification through manifold vertex clustering.

The error-constrained *k*-d tree grid simplification (i.e., the simplification of the *k*-d tree via error metrics) is initiated by successively removing minimal partitioning faces from the grid. The simplification process is terminated if the computed error exceeds a user-defined threshold. The intersection points and normals for the *k*-d tree's working grid edges are represented by a set of planes, determining the primitive shape representing the isosurface of the rendered object [10].

The octree simplification using QEFs is a geometry simplification scheme introduced as part of the standard dual contouring algorithm. The QEF for each corresponding leaf is computed as given by Equation (1) (Ju et al. [20]):

$$E[x] = \sum_i (n_i \cdot (x - p_i))^2, \tag{1}$$

where n_i and p_i refer to the resulting intersections (including unit normals) between the contour and edges of the cube. The QEFs are computed at the interior nodes of the octree by combining these QEFs with their corresponding leaves. Nodes whose QEFs have a residual smaller than a given threshold are collapsed into leaves, thereby simplifying the octree.

The octree-based simplification through manifold vertex clustering involves constructing a vertex tree for vertices at the finest level of the octree and determining the QEF for each clustered vertex. To simplify the octree, only vertices topologically related to the surface are clustered together. Each clustered vertex must satisfy an additional topology criterion to ensure it represents a simplified contour that is two-manifold, referred to as *the manifold criterion* [21]:

$$X(S_v) = V(S_v) - E(S_v) + F(S_v), \tag{2}$$

where $X(S_v)$ is the Euler characteristic, $V(S_v)$ is the total number of vertices, $E(S_v)$ is the total number of edges, and $F(S_v)$ is the total number of faces.

2.1.3 Metrics used in topology simplification schemes

Dual contouring based simplification of isosurfaces involves the representation and minimization of a relevant QEF to achieve accurate results. Equation (1) in 2.1.2 is constructed through a collection of intersection points p_i and their normals n_i. The function $E[x]$ can be considered as the inner product $(Ax-b)^T(Ax-b)$, where A is a matrix whose rows are the normals n_i, and b is a vector resulting from $n_i \cdot p_i$. The QEF can be expanded as shown in Equation (3):

$$E[x] = x^T A^T Ax - 2x^T A^T b + b^T b, \tag{3}$$

where the matrix $A^T A$ is a symmetric 3×3 matrix, $A^T b$ represents a three-length column vector, and $b^T b$ is a scalar. Consequently, only the matrices $A^T b$, $A^T A$, and $b^T b$ need to be stored in 10 floats for optimization, rather than storing the matrices A and b. The minimization value $\hat{x}$ required for $E[x]$ can be obtained by solving the normal equation (4):

$$A^T A\hat{x} = A^T b. \tag{4}$$

Another application of QEFs in dual contouring algorithms is for computing error metrics. However, it differs from the method used for *k*-d tree error-constrained simplification of isosurfaces. Although both approaches calculate simplification errors through matrix manipulation of intersection points and Euclidean distances, the computation of the simplification error E in the *k*-d tree method is based on the position of a set of points relative to the simplified object, rather than on a minimization value [22]. The error between the simplified and original representation can be computed by taking the least Euclidean lengths $d_S(v)$ between the set of points $v \in P_{\text{on}}$ on the surface representation of the simplified isosurface, and the lengths $d_I(P_{\text{in}})$ and $d_O(P_{\text{out}})$ from points $p_{\text{in}} \in P_{\text{in}}$ and $p_{\text{out}} \in P_{\text{out}}$ on the exit of the inner and outer of the simplified object. These error metrics can then be categorized as *in-metric*, *on-metric*, and *out-metric*. The on-metric error E_{on} is defined from the distances $d_S(v)$ and measures the distance between the original and simplified isosurface representations. The in-metric E_{in} and out-metric E_{out} errors are established from the relevant distances $d_I(P_{\text{in}})$ and $d_O(P_{\text{out}})$, facilitating the computation of the error through topological modifications. These metrics can be defined for user-specified error thresholds ϵ_{on}, ϵ_{in}, and ϵ_{out} as

$$E_{\text{on}} \leq \epsilon_{\text{on}}, \quad E_{\text{in}} \leq \epsilon_{\text{in}}, and \quad E_{\text{out}} \leq \epsilon_{\text{out}}. \tag{5}$$

Intersection points and normals for the *k*-d tree's working grid edges can be represented by a set of planes S, determining the primitive shape representing the isosurface. To compute the distance of a point relative to the isosurface, the corresponding planes closest to the polygon must be identified at each completion of the isosurface extraction.

Suppose a plane $s \in S$ consists of a set of points P_s nearest and sampled from the associated polygon P_{on}. The on-metric is defined as

$$E_{\text{on}} = \max_{s \in S} \frac{1}{|P_s|} \sum_{v \in P_s} d_S(v)^2, \tag{6}$$

where $d_S(v)$ represents the distance of a point v from the plane s. Let I denote the exit of the inside of an entity according to the k-d tree grid representation, with O as the exit of the object's exterior. The in-metric and out-metric can be expressed as follows:

$$E_{\text{in}} = \max_{p_{\text{in}} \in P_{\text{in}}} d_I(p_{\text{in}}), \tag{7}$$

$$E_{\text{out}} = \max_{p_{\text{out}} \in P_{\text{out}}} d_O(p_{\text{out}}). \tag{8}$$

Given that the length d_I from p_{in} to the exit of the inside of the object I correlates with the distance to the edge of the surface S when p_{in} and zero otherwise, the equations can be rewritten as follows:

$$E_{\text{in}} = \max_{p_{\text{in}} \in I P_{\text{in}}} d_S(p_{\text{in}}), \tag{9}$$

$$E_{\text{out}} = \max_{p_{\text{out}} \in O P_{\text{out}}} d_S(p_{\text{out}}). \tag{10}$$

In a manner similar to the minimization and representation of QEF, the exact evaluation of finding the value of the on-metric depends on finding the summation of squared distances $E(n, d)$ among points $v_1, \ldots, v_k \in P_{\text{on}}$ and the plane $s \in S$ as defined by a normal n along with a scalar d. This can be performed with an equation similar to the general QEF equation (1):

$$E(n, d) = \frac{1}{k} \sum_{i=1}^{k} (n^T v_i + d)^2. \tag{11}$$

By considering the following coefficients,

$$A = \sum_{i=1}^{k} v_i v_i^T, \quad b = \sum_{i=1}^{k} v_i, \quad c = k, \tag{12}$$

where A represents a symmetric 3×3 matrix, b is a 3-vector, and c is a scalar, Equation (11) can be further expressed as:

$$E(n, d) = \frac{1}{k}(n^T A n - 2 d n^T b + d^2 c). \tag{13}$$

Given $P = (A, b, c)$, which is referred to as *the dual quadric*, and assuming the plane passes through the mean of points v_i, then the quantity is $d = -\frac{n^T b}{c}$. The corresponding normal n is associated with the eigenvector of the covariance matrix Z made of points v_i, which corresponds with the least eigenvalue. According to the dual quadric, Z can be computed as follows:

$$Z = \frac{1}{k-1} \left(A - \frac{b b^T}{c} \right). \tag{14}$$

2.2 Related work

Isosurface extraction refers to a broad range of techniques that can be used in the visualization of three-dimensional scalar data [23]. The literature review provides an analysis of two of the most common techniques, namely the *marching cubes* and *dual contouring* algorithms, and their combination, the *dual marching cubes* algorithm. The review also discusses the various motivations behind further development of each technique.

2.2.1 The marching cubes algorithm and its modifications

The foundational marching cubes (MC) algorithm of isosurface extraction for scientific visualization of high resolution 3D volumetric data was developed by Lorensen and Cline in 1987 [9]. The MC algorithm took as input a uniform grid whose vertices were samples of the function *f(x, y, z)* and extracted a surface as the zero-contour. For each cube in the grid, the MC algorithm examined the values at the eight corners of the cube and determined the intersection of the surface with the edges of the cube. A look-up table indexed by the sign configuration at the eight corners that yielded the topology of the surface in side of that cube was provided. Processing each cube in the grid resulted in the complete surface. There have been numerous modifications of this reference algorithm aiming at its further improvement and/or simplification. An extensive survey of the marching cubes algorithms looking at the evolution of the standard MC algorithm from 1987 to 2006 can be found in [24].

Rajon and Bolch [25] considered the original MC algorithm and several of its modifications available at the time of their research, as well as the issues identified in each reviewed MC-based alrorithm. Their specific interest was to generate isosurfaces delineating tissue interfaces from the gray-level medical (trabecular bone) images. Rajon and Bolch indicated that the rectangular shape of image voxels generated voxel effects that altered the outcome of isosurface generation. To minimize voxel effects, Rajon and Bolch proposed an adaptation of the MC algorithm in which a trilinear interpolation of the gray levels was used to generate a hyperboloid surface. According to Rajon and Bolch, the adapted technique was capable of solving the ambiguity problem of the original MC algorithm. It also allowed to reduce the data size inherent to the triangulated surface and provided a simplified algorithm to accurately measure distances within the image. The trilinear interpolation method removed voxel effects and produced chord-length distributions across image regions.

Maple [26] suggested an application of the original MC algorithm and its two-dimensional variation (*marching squares algorithm*) for geometric design and space planning. The proposed addition to the MC algorithm allowed to approximate the area or volume of the object. This method could also be used in estimation of the area encapsulated between two points on the surface and a line or the volume encapsulated between three points on the surface and a plane.

Research by Andújar et al. [27] focused on improving the efficiency of the MC algorithm. The proposed solution was to select one valid topology that would minimize a desired topological or combinatorial measures (i.e., the total triangle count, the number of connected shells, or the total genus). Since the measures to be minimized in this research were not affected by the placement of the vertices of the isosurface mesh, Andújar et al. suggested to position each vertex (i.e., an angular point) of the isosurface at the midpoints of the lattice edges joining inside and outside samples. Moving vertices to more appropriate locations along their lattice edges and, therefore, preserving the topological and combinatorial properties of the isosurface led to a more efficient algorithm.

Lee et al. [11] proposed *the modified marching cubes (MMC)* algorithm by reconsidering the triangulated cubes of the original MC algorithm. The MMC algorithm allowed for regularization of the connectivity of the isosurface mesh. Consequently, a maximum valence of six vertices was achieved in the modified algorithm. The MMC algorithm demonstrated an improvement of the mesh topology and a significant reduction in the connectivity coding cost. The algorithm utilized an approach whereby a simple remark was based on how the isosurface intersected its associated height voxel cube.

Almost all methods based on MC utilize a look-up table to triangulate the isosurface. Renbo et al. [28] presented a variation of the MC algorithm that did not use a conventional look-up table. Instead, it relied on an automated triangulation strategy based on critical points lying on the isosurface in the interior of the cube. These critical points were classified as face shoulder points, body shoulder points, and inflection points. According to Renbo et al., for any case of cube configuration, the improved algorithm was capable of generating a topologically accurate approximation to the isosurface of the trilinear interpolant within the cube. The accuracy of the reconstruction process, as compared to the original MC algorithm, was improved using some extra points located on the characteristic positions of isosurface.

Jin et al. [12] identified disadvantages in the original MC algorithm that came from the use of surface configurations of cubes leading to the wrong surface production and hole generation. In response to these weaknesses, the authors proposed an improvement of the MC method by reassigning zero to vertices in the surface and one to vertices on the surface, and then redefining 15 typical configurations considered in the marching cubes method. They reassigned zero to vertices in the surface and one to vertices on the surface, and then improved the marching cubes algorithm by redefining the configurations for the fifteen cases of configurations considered in the original marching cubes method. To evaluate the performance of the proposed improved marching cubes method, Jin et al. reconstructed the surface for MRI volume data using the proposed method along with the original MC method to demonstrate improvement.

Strand and Stelldinger [13] presented three adaptations of the original MC algorithm for preserving the topology of the marching cube producing different local configurations on a face-centered cubic (FCC) grid. The first adaptation presented a combination of the three partitioning schemes employed in earlier MC algorithms (i.e., a Delaunay mesh partitioning [29]). Another adaptation involved tetrahedra mesh partitioning of the FCC grid and possible local configurations resulting in a least simplified instance of topology preserved MC. The third adaptation used rhombic dodecahedra partitioning and demonstrated the best simplification results out of three implementations.

Cui and Liu [30] created *the simplified marching cubes (SpMC)* algorithm as a modification of the standard MC algorithm. They suggested a change in the position of the isosurface extracted vertex at an interpolation point on a cube edge which corresponded to the cube vertex. The modified algorithm required fewer triangulation cases and, in most cases, fewer extracted triangles than the original MC algorithm. The SpMC algorithm did not need interpolation calculations of vertex positions and normal vectors required in the original MC algorithm.

Research by Etiene et al. [31] did not aim at improving the original MC algorithm or any of its variations. Instead, they presented a tool for selection of the most appropriate algorithm among MC variations. Etiene et al. presented techniques for assessment of the behavior of isosurface extraction codes and verification of various visualization algorithms. These techniques were used to distinguish whether anomalies in isosurface features could be attributed to the underlying physical process or to artifacts from the extraction process. Etiene et al. argued the necessity of ”verifiable visualization” - subjecting visualization algorithms (including MC-based ones) to the same verification process as in any other academic discipline. The focus of the verification was on topological properties of isosurface extraction algorithms. Etiene et al. derived formulas for the expected order of accuracy (or convergence rate) of several isosurface features, and compared them to experimentally observed results in the selected codes. According to Etiene et al., results of the verification of various algorithms (i.e., the MC algorithm and its variants) could assist with the selection of the appropriate isosurface extraction technique and visualization algorithm.

Xu et al. [14] suggested three enhancements of the original MC algorithm. The first enhancement was to replace the cube edge linear interpolation with a midpoint selection scheme. The triangular patches generated in this way led to a smoother isosurface in local area thus contributing to the mesh simplification. The second improvement was the use of a three-dimensional array to store the coordinates of the points of intersection between the cube edge and the isosurface. This helped to avoid repeated calculation of these points. The third enhancement was to employ the edge contraction method to reduce the number of triangular patches. When two vertices satisfied the stated constraint conditions, the two vertices would merge into one, and the edge consisting of the two vertices would contract into a point. If the conditions were not met, the triangle would be kept.

Vignoles et al. [15] suggested a triangulation method leading to the development of *the simplified marching cubes (SMC)* algorithm. The SMC algorithm relied on a mesh built with vertices linked together into triangles of a cuberille grid. Triangles were composed of vertices of the cuberille grid on which the processed data is defined. In the cuberille model, an object is typically represented as a collection of cube shaped voxels [32]. Meshes were *manifold*, i.e., topologically consistent and without holes. The accuracy of the obtained meshes was reported to be lower than in the original MC algorithm, but higher than in the cuberille approximation. The algorithm produced a configuration with a reduced number of triangles as a result of a reduction in the number of vertex points.

The modification by Du et al. [16] focused mainly on improving efficiency and topological accuracy of the original algorithm. To address the issue of efficiency, they altered the way in which an isosurface intersected the hexahedral voxels. The isosurface usually only intersects some voxels. The standard MC algorithm checked and computed all hexahedron voxels in the three dimensional data area. This operation required significant time and resulted in the low efficiency. In the proposed solution, if the hexahedron voxels intersect with the isosurface, the isosurface would be in the continuity along the six surfaces of the hexahedron. If a cube intersects with the isosurface, there are intersection lines on some surfaces of the six surfaces on the cube. The surfaces of the adjacent cube (front, back, upper, lower, left and right) would be extended according to a certain order. Du et al. proved that 90% of the isosurfaces were composed of these six cases. It helped to significantly reduce the time and, therefore, improve efficiency. The issue of topological accuracy was addressed in the following way. In the original MC algorithm, the nodal between the isosurfaces and the voxel boundary was computed on the assumption that the function value changed linearly along the voxel boundary. When the density is high in the three dimensional discrete data area, i.e., when the voxels are very small, this assumption is true. However, if the voxels are large in sparse data area then the isosurfaces could not be accurately obtain. Therefore, the recommendation was to use high data field density and small voxels, so the model would be reconstructed in the high accuracy and exact structure.

Custodio et al. [33] developed *the corrected marching cubes 33 (C-MC33)* algorithm addressing topological issues of the marching cubes 33 algorithm (MC33). The original marching cubes 33 algorithm was, in its turn, a modification of the classical MC algorithm proposed by Tcherniaev [34] and further extended by Lewiner et al. [35]. The MC33 algorithm was one of the first modifications of the MC algorithms aiming to preserve the topology of the trilinear interpolant. The main contribution of Tcherniaev was that the original MC triangulation table was extended to 33 cases – hence, the name of the algorithm. Custodio et al. found several issues in the MC33 algorithm by Tcherniaev and its later extension by Lewiner et al. The identified issues included (1) disambiguation in Cases 10, 12 and 13.5 of the 33 cases considered by Tcherniaev [34], (2) non-manifold surfaces [34], and (3) failure to compute cut plane heights [35]. Custodio et al. presented solutions for the identified issues and implemented them into C-MC33, a more topologically correct version of MC33 [33].

The focus of the study by Chen and Jin [36] was on utilization of a graphics processing unit (GPU) for parallel optimization of polygonized isosurfaces in the MC algorithm. Chen and Jin introduced a GPU-based approach to polygonize and optimize isosurface meshes for implicit surfaces. Specifically, Chen and Jin designed schemes to exploit the parallel features of the GPU hardware by optimising both the geometry (vertex position, vertex distribution, triangle shape, and triangle normal) and the topology (connectivity) aspects of a mesh. According to the authors, this method demonstrated improvement on the resultant mesh quality and acceleration of the isosurface extraction process as compared with CPU-based approach.

Athawale et al. [37] did not propose a modification of the MC algorithm but investigated the impact of data uncertainty on topology and geometry extraction in MC algorithms. They proposed an edge-crossing probability based approach to predict underlying isosurface topology for uncertain data. Athawale et al. pointed out that data uncertainty, characterized by probability distributions, could be propagated through the isosurface extraction process. They derived a probabilistic version of the midpoint decider that, according to the obtained results, resolved ambiguities that arised in identifying topological configurations. Athawale et al. designed a probabilistic techniques for handling uncertainty in cell configurations for isosurface topology etermination. They proposed vertex-based classification and edge-based classification methods to classify vertex signs. The obtained results demonstrated the advantage of nonlocal statistics approach for characterising data uncertainty over locally estimated parametric and non-parametric densities.

Most of the modified MC algorithms used a single 3D scalar field for rendering isosurfaces but did not take into account the characteristics of the scalar field itself. Ronghuan et al. [38] proposed a isosurface extraction method based on multi-resolution scalar field construction and seamless intersecting surface. Research by Ronghuan et al. aimed at improving the efficiency of isosurface extraction and rendering by reducing the isosurface extraction data while maintaining the effect of isosurface rendering. The multi-resolution scalar field was constructed to address the problems of ambiguity and gaps generated between different (low and high) resolution scalar fields of isosurfaces.

A splicing of contours between scalar fields with different resolutions was performed in order to remove the ambiguities and gaps. Thus, the contours of the high-resolution scalar field were kept uniform, and the extraction of low-resolution scalar fields was undertaken with unique triangular patches.

2.2.2 The dual contouring algorithm and its modifications

Dual contouring (DC) was first proposed by Ju et al. in 2002 [20]. Dual contouring is a feature-preserving isosurfacing method that extracts crack-free surfaces from both uniform and adaptive octree grids [20]. Dual contouring aims at reconstructing even sharp features of an object with the help of the gradient of the scalar field. Dual contouring requires that a scalar value and a gradient vector be assigned to each vertex of the octants. For this reason, this algorithm cannot be used for volume fraction data where only a scalar value is assigned to each octant. Dual contouring is a method used for extracting the surface boundary of an implicit volume. The method is *dual* in the sense that vertices generated by DC are topologically dual to faces in the MC algorithm. In dual contouring, a uniform grid is superimposed on the implicit volume. The grid cubes are represented as nodes in an octree data structure. The advantage of DC is that it can reproduce sharp features by inserting vertices anywhere inside the grid cube, as opposed to the marching cubes (MC) algorithm that can insert vertices only on the grid edges.

Greß and Klein[17] proposed the utilization of a *k*-d tree-based hierarchy for an implicit object representation. They asserted that the *k*-d tree-based hierarchy was superior to the octree in terms of adapting to the object surface. Greß and Klein obtained more compact implicit representations especially in case of thin object structures. They described a new isosurface extraction algorithm based on *k*-d-trees instead of octrees in the original dual contouring algorithm by Ju et al. [20]. The process resulted in generation of two-manifold meshes even for *k*-d trees with cells containing multiple surface components. In addition, a simplification framework was created for the surfaces represented by the *k*-d tree based on quadric error metrics. The framework allowed controlled topological simplification of the object.

Ju et al. developed a method for contouring a signed grid whose edges were tagged by Hermite data (i.e., exact intersection points and normals) [20]. This method did not require to explicitly identify and process features as in previous Hermite contouring methods. Using a numerically stable representation for QEFs, Ju et al. developed an octree-based contouring method. The DC method imposed no constraints on the octree and did not require crack patching.The important feature of the original DC algorithm by Ju et al. is that it adopted an octree as the data structure to adaptively represent extracted isosurfaces.

The original DC algorithm over the years underwent various modifications aiming to improve its performance. For example, Zhang et al. proposed a modified dual contouring algorithm DC with topology-preserving simplification [39]. The aim of the suggested modification was to preserve the disconnected surface components in cells during isosurface simplification. Zhang et al. represented isosurface components in a form of *enhanced cells*. In an enhanced cell, each surface component was represented by a vertex and its connectivity information. A topology-preserving vertex clustering algorithm was applied to build a vertex octree. An enhanced dual contouring algorithm was employed to extract error-bounded multi-resolution isosurfaces from the vertex octree while preserving the finest resolution isosurface topology. A connectivity-guided vertex clustering algorithm was used to simplify the isosurface components. After building a hierarchically clustered vertex octree, topology-preserved isosurfaces could be extracted under various error bounds by the enhanced dual contouring algorithm. Dual contouring with topology-preserving simplification using enhanced cell representation.

Schaefer et al. [40] modified their original DC algorithm in order to guarantee that the mesh generated was *manifold* even under adaptive simplification. They extended the original DC algorithm by complementing it with an octree-based topology-preserving vertex-clustering algorithm for adaptive contouring. The contoured surface generated by the extended method contained only manifold vertices and edges, preserved sharp features, and possessed better adaptivity than the original algorithm.

Zhang and Qian [41] developed a modification of the octree-based dual contouring (DC) method to construct surface and volumetric meshes for complicated domains. After considering all possible topology configurations, they developed an extension of the standard DC algorithm which, according to them, guaranteed the correct topology. The process of constructing surface and volumetric meshes included the following steps. First, one base mesh was generated using the original DC method. Then all the octree leaf cells were considered and categorized into 31 topology groups. In order to discriminate between these cells, the values of their face and body saddle points were computed based on a trilinear representation inside the cells. Then Zhang and Qian modified the base mesh and introduced more minimizer points within the same cell. With these minimizer points the mesh connectivities were updated to preserve the correct topology. This method was further extended to 3D tetrahedral mesh generation via an advancing front technique. A Laplacian smoothing technique was applied to improve the mesh quality; for tetrahedral mesh a combination of edge-contraction, smoothing and optimization was also applied. In a more recent work, Liang and Zhang [42] extended initial research by introducing the modification of the DC algorithm for adaptive triangular or tetrahedral mesh generation to guarantee a better angle range. The algorithm was based on a quadtree or octree structure, and could generate interior and exterior meshes with conformal boundary. The results demonstrated that the improved octree-based dual contouring method was capable of generating guaranteed-quality meshes.

Peixoto and de Moura [43] complemented the original DC algorithm with two discretization methods – *the non-compact dual simplification (NDS)* and *the sewing octree*. These discretization methods were developed to operate on polygonal surfaces specifically generated by the octree-based adaptive dual contouring algorithm. The NDS method focused on preserving the simplified topology of non-compact surfaces. However, this method was also applicable to compact surfaces if their polygonalization regions did not have any regions of non-compact surfaces. The sewing octree scheme provided a way of combining two or more octrees that shared faces or edges and contained portions of the surface polygonalized with dual contouring. These methods could be employed either independently or coupled by (1) dividing the original cube in two or more cubes, (2) carrying out the polygonalization, (3) simplifying these regions with NDS, if necessary, and (4) glueing the resulting surfaces with the sewing octree. This procedure, as authors stated, guaranteed that the resulting surface would have the same topology as the original surface.

Rashid et al. [44] suggested an improvement of the original DC method to produce watertight and two-manifold surface meshes. They observed that the original DC method produced only one vertex for each grid cube and, therefore, was unable to generate watertight and non-manifold meshes. The solution was to decompose an ambiguous grid cube into a maximum of 12 tetrahedral cells. In addition to that, Rashid et al. also introduced the polygon generation rules. The improved algorithm resulted in the production of watertight and two-manifold surface meshes.

Varadhan et al. [45] presented two algorithms for accurate polygonization of implicit surfaces from volumetric data, namely, feature-sensitive adaptive subdivision and isosurface reconstruction. Isosurface reconstruction used directed distances, i.e., distance along a direction, to perform an exact edge intersection test. This edge intersection test was used to detect intersections of the edge with a surface. This test was combined with the original dual contouring algorithm to obtain an improved reconstruction algorithm which Varadhan et al. called *extended dual contouring*. It was capable of reconstructing thin features while avoiding creation of additional handles. The algorithm tool into account the characteristics of the grid and considered complex edges. It enumerated all the intersections along the edges, separated them into components and reconstructed the isosurface locally within each cell.

2.2.3 Dual marching cubes

Dual marching cubes (DMC) is a combination of the MC and DC methods, and can be considered as a modification of both. However, due to its significant impact on the visualization discipline, it is considered separately from other MC and DC modifications. Schaefer and Warren [10] presented a method for contouring an implicit function using a grid topologically dual to structured grids such as octrees. First, this algorithm computed an octree from the input scalar field and then generated an appropriate dual grid based on the scalar field. After the generation of the dual grid, mesh vertices are computed in the same manner as in the original MC algorithm. Because the dual marching cubes algorithm extracted a surface from a grid structure whose cells were assigned values, it was applicable to volume fraction data on an octree. By aligning the vertices of the dual grid with the features of the implicit function, the proposed algorithm was able to reproduce thin features of the extracted surface without the excessive sub-division required by the original MC and DC methods. The DMC method led to a crack-free, adaptive polygonalization of the surface that reproduced sharp features.

However, one of the main issues with the DMC method by Schaefer and Warren was related to the topology of the extracted surfaces. A data structure other than the uniform grids had to be employed for efficient and accurate surface representations of the generated implicit models. The uniform grids could not provide grid sizes with enough space for extracting implicit models. Another limitation of the DMC method is the lack of sharp features from the environments generated by the previous implementations of the MC algorithms. This problem was attributed to improper contouring as large flat regions would often get tiled with small polygons resulting into the generated meshes.

In the same year, but a few months later than Schaefer and Warren [10], Nielson [46] proposed a similar algorithm coincidentally named *dual marching cubes*. The dual marching cubes algorithm presented by Nielson was, however, a different strategy to reconstruct an isosurface from the volume data. The intersection of the isosurface with the cell was approximated by a polygon on the cell faces. Nielson's work included the definition and computational algorithms for a new class of surfaces which were *dual* to the isosurface produced by the MC algorithm. These isosurfaces had the same separating properties as the MC surfaces but were comprised of quad patches eliminating the common negative aspect of poorly shaped triangles of the MC isosurfaces. Based upon the concept of *the dual operator*, Nielson proposed an iterative scheme for generating smooth separating surfaces for binary, enumerated volumes often produced by segmentation algorithms. Importantly, the DMC method by Nielson was not based on QEF or octrees. The primary distinction between the DMC algorithm by Schaefer and Warren [10] and the one by Nielson [46] lied in how the extracted vertex locations were determined. Schaefer and Warren defined the vertex locations as the minimizer of a QEF, whereas Nielson specified them based on a geometric function of the primal mesh geometry, such as the face centroid.

A series of further modifications of the two original DMC methods were carried out over the years. The DMC algorithm by Schaefer and Warren was modified by Kim et al. [47] by introducing an interpolation scheme on the octree volume fraction data. The algorithm was similar to the original dual marching cubes except for the computation of the mesh vertex. The mesh vertices were linked together to create the surface mesh. An approximation based on the shape of the octants of a sphere was proposed to achieve an accurate vertex computation of the octree.

Greß and Klein [17] used a different data structure to represent implicit objects – the *k*-d tree. The *k*-d tree grid was built on the assumption that an appropriate grid partitioning criterion for subdividing splitting planes was determined. Construction of the grid took a top-down routine as follows. For every subdivision step, the active grid edges had to be detected and their related intersection information was computed. However, to achieve two-manifold and topologically correct representations, the subdivision scheme maintained a list of faces in the cell. This allowed faces contained in the list to be clipped against the splitting plane throughout each subdivision step, and these faces were handed over to the corresponding sub-cells after being clipped. After generating each vertex during the clipping process, a record was made of the axes corresponding to the planes clipped against them. Even though the *k*-d tree was selected in [17] to efficiently replace the octree, the study did not compare the efficiency of these structures across various simplification error thresholds. This gap in comparison motivated the work presented in this paper.

Grosso and Zint [48] suggested a modification of the Nielson's DMC algorithm. The proposed method aimed at reconstructing surfaces from volume data using a dual marching cubes approach without look-up tables (similar to the Nielson's method). The method generated quad-only meshes which were consistent across cell borders, i.e., they were manifold and watertight. Vertices were positioned exactly on the reconstructed surface leading to high accuracy. A half-edge data structure was used for storing the meshes for further processing. The method processed elements in parallel and, therefore, ran efficiently on GPU. Due to the transition between layers in volume data, meshes had numerous vertices with valence of three. Grosso and Zint used simplification patterns for eliminating quads containing these vertices reducing the number of elements and increasing quality.

He et al. [49] considered both DMC algorithms but opted for modification of Nielson's DMC algorithm by adding degrees of freedom to flexibly position each extracted vertex within its dual cell. This work proposed gradient-based mesh optimization of 3D surface mesh by representing it as the isosurface of a scalar field. According to He et al., the existing implementations were designed to extract meshes from fixed, known fields, and in the optimization setting; therefore, they lacked the degrees of freedom to represent high-quality feature-preserving meshes, or demonstrated numerical instabilities. He at al. proposed an isosurface representation aiming at optimization of an unknown mesh with respect to geometric, visual, or even physical objectives. The authors introduced additional parameters into the representation allowing for local adjustments to the extracted mesh geometry and connectivity. These parameters were updated along with the underlying scalar field via automatic differentiation when optimising for a downstream task. The extraction scheme presented extensions to optionally generate tetrahedral and hierarchically-adaptive meshes.

2.3 Literature review summary

A critical review of the marching cubes, dual contouring and dual marching cubes algorithms and their modifications prepared us to the implementation of an approach for evaluating the efficiency of the *k*-d trees and octrees in the simplification of isosurfaces in a dual contouring algorithm for a specific or given error. The problem addressed in this research is the lack of comparative tools for the developers to guide the selection of adaptive data structures (i.e., k-d trees and octrees) during implementations.

Table 1 and Table 2 presents a summary of the key findings of the reviewed literature given in a chronological order.

3 Proposed approach to implementation of the two algorithms

The *k*-d tree- and octree-based manifold dual contouring algorithms for isosurface extraction were implemented with the objective of comparing their efficiency. The implementation of both algorithms was performed in two phases: construction of the underlying rectilinear grid (Phase 1) and isosurface extraction of implicit models through error-controlled simplification (Phase 2). Within each phase there were several sub-phases. The construction of the underlying rectilinear grid included the following sub-phases: formulation of QEFs (Sub-phase 1.1), generation of the scalar field (Sub-phase 1.2), composition of vertex data with Hermite data (Sub-phase 1.3), preparation of the manifold criterion topology preserving scheme for contouring the scalar field (Sub-phase 1.4), and construction of the general mesh in accordance with the topology preserving scheme (Sub-phase 1.5). The isosurface extraction of implicit models through error-controlled simplification included the following sub-phases: generation of the scalar field of the implicit surface representation (Sub-phase 2.1), preparation of the computation of topology preserving vertex clustering using QEFs (Sub-phase 2.2), and QEF controlled isosurface simplification on implicit *k*-d tree- or octree-based adaptive grids (Sub-phase 2.3). A detailed flowchart representing each phase and sub-phase in the implementation of the *k*-d tree algorithm is depicted in Figure 3.

The flowchart for the original octree algorithm is not included in this paper as it was implemented following prior research by Schaefer et al. [40]. The implementation involved the standard process of building an octree hierarchy

Table 1: The summary of the reviewed literature (in the chronological order) – Part I

Techniques	Key Results	Identified Gaps
Associative k-d trees [18]	Multi-dimensional binary search trees	No comparison with octrees
Octree generation [19]	Efficient volume decomposition	Limited to octree structures
Marching cubes [9]	High-resolution 3D surface construction	Inconsistencies in visual depictions
Dual contouring of Hermite data [20]	Crack-free, gradient-based surfaces	Requires scalar and gradient data
Adaptive isosurface k-d Trees [17]	Efficient representation of thin objects	Lack of comparative analysis with octrees
Trilinear interpolation adaptation [25]	Minimized voxel effects	Voxel shape impacts outcome
Feature-sensitive subdivision and isosurface reconstruction [45]	Reconstructed thin features, avoided additional handles	Limited to specific grid structures
Dual marching cubes (based on QEFs) [10]	Preserved sharp features, adaptive contouring	Topological inaccuracies
Dual marching cubes (based on geometric centroid) [46]	Smoother, quad-dominant meshes without look-up tables	Did not use QEF minimization, limited empirical comparison
Optimizing topological and combinatorial complexity [27]	Efficient isosurface extraction	Limited focus on topology
Regularized marching cubes [11]	Improved mesh topology, maximum vertex valence of six	Did not address volumetric data complexity
Improved marching cubes [12]	Addressed wrong surface production	Hole generation in original method
Topology preserving marching cubes [13]	Preserved topology on face-centered cubic grid	Limited to specific grid structures
Improved marching cubes [14]	Smoother isosurfaces, reduced triangle patches	No comprehensive performance evaluation
Simplified marching cubes [15]	Efficient discretization for deposition/ablation simulations	Lower accuracy compared to original MC algorithm
Efficient improved marching cubes [16]	Increased efficiency and topological accuracy	Focused on specific efficiency improvements

and performing dual contouring with manifold vertex clustering to ensure topological correctness and detail preservation. The readers are advised to refer to the original publication [40] for a more detailed description of the algorithmic steps for the key phases. Both implemented manifold dual contouring algorithms used rectilinear grids to represent the extraction of implicit models. Implementation by Greß and Klein [17] employed the combination of uniform grids with the topology preserving scheme for manifold extracted isosurfaces with sharp features. This implementation resulted in the generation of too many polygons. This was not desirable, particularly during extraction of relatively flat surfaces. Numerous polygons would make it difficult to render large models made of flat surfaces.

3.1 Phase 1: Construction of the underlying rectilinear grid

Phase 1 of the implementation of the two algorithms involved constructing the underlying rectilinear grid. The advantages of rectilinear grids are a negligible memory footprint and readily support smooth data reconstruction, though with reduced geometric flexibility [50]. Construction of the rectilinear grid was required for working with implicit surface representations of extracted isosurfaces.

Sub-phase 1.1 involved the formulation of the QEFs. This study adopted the QEF formulation from Ju et al. [20] as given by Equations (1) and (3) in Subsection 2.1.2. These equations allowed us to determine the corresponding vertex locations and the error associated with those locations. In a situation where the accuracy of positioning the vertices is more important than the basic representation of the QEF, the relationship between the orthogonal matrix Q of the QR

Table 2: The summary of the reviewed literature (in the chronological order) – Part II

Techniques	Key Results	Identified Gaps
Corrected marching cubes 33 [33]	Addressed topological issues in MC33	Hole generation in original method
GPU-based polygonization and optimization [36]	Improved mesh quality and extraction speed	Limited to GPU-based methods
Edge-crossing probability for uncertainty handling [37]	Resolved ambiguities in topology	Complexity in handling uncertainty
Gradient-based mesh optimization [49]	Flexible vertex positioning, high-quality meshes	Numerical instabilities, limited flexibility
Improved dual contouring [44]	Watertight and two-manifold meshes	Complexity in implementation
Interpolation scheme on octree volume fraction data [47]	Accurate vertex computation, improved mesh quality	Limited to octree structures
Multi-resolution isosurface extraction [38]	Efficient data reduction while maintaining rendering	Difficulty in accurately determining the isosurface location, holes that may occur when merging or transitioning between isosurfaces
Parallel dual marching cubes without look-up tables [48]	High accuracy, efficient on GPU	Transition issues between volume data layers

decomposition can be used to satisfy the relation $Q^TQ = I$, leading to the error equation $E[x]$ to be determined through Equation (15):

$$\begin{aligned}(Ax-b)^T(Ax-b) &= (Ax-b)^TQ^TQ(Ax-b)\\ &= (QAx-Qb)^T(QAx-Qb) \quad (15)\\ &= (\hat{A}\hat{x}-\hat{b})^T(\hat{A}\hat{x}-\hat{b})+r^2.\end{aligned}$$

In this form, the evaluation of $E[x]$ becomes attainable by computing the squared vector product of the vector $\hat{A}x - b$, before adding r^2. For a non-singular $\hat{A}$ case, minimization of x can be calculated through solving the equation $\hat{A}\hat{x} = \hat{b}$ by performing back substitution. $\hat{A}$ is calculated from the noisy normals tending to always be coplanar. The same matrix $\hat{A}$ becomes approximately singular. Minimizing the $\hat{x}$ value results in setting in the far outside of the outlining cube. Solving this problem means that the singular value decomposition (SVD) of $\hat{A}$ must be computed, and its pseudo-inverse must be formed by cutting down its trivial singular values:

$$\hat{A} = U\Sigma V^T, \quad (16)$$

where Σ is a diagonal matrix with singular values. The pseudo-inverse $\hat{A}^+$ of $\hat{A}$ is given by Equation (17)

$$\hat{A}^+ = V\Sigma^+U^T, \quad (17)$$

where Σ^+ contains reciprocals of non-zero singular values. Using SVD, the minimizer x is computed as follows:

$$x = V\Sigma^+U^Tb. \quad (18)$$

Sub-phase 1.2 involved the generation of the scalar field. A scalar field refers to a function ϕ which specifies a constant or scalar value to every point in three-dimensions. A scalar field is a function associating a single number to every point in a space. The focus of this implementation was on isosurface extraction of implicit surface representations, and the scalar fields were generated from implicit functions.

The formulation of QEFs and generation of the scalar field in Sub-phases 1.1 and 1.2 provided the inputs for Sub-phase 1.3, namely, composition of vertex data with Hermite data. Inputs included information regarding the positioning of the vertices and their related attributes. A scalar field represents every angular point in the three-dimensional space referred to as a vertex. Each of these vertices contains attributes such as a vertex position and normal. This information was required in the extraction of the isosurface for the algorithm to determine what type of surface was extracted at that particular vertex. The approach used to process vertices tagged with Hermite data was similar to the one introduced in [20].

Sub-phase 1.4 involved the preparation of the manifold criterion topology preserving scheme. For the implementation involving isosurface simplification using octrees, the manifold vertex clustering technique was adopted from Schaefer et al. [40]. For the implementation using k-d trees in the manifold vertex clustering technique by Schaefer et al., the octree isosurface simplification was replaced with the k-d tree adaptive isosurface simplification. For both implementations the simplification error thresholds were the same.

Sub-phase 1.5 pertained to the construction of the gen-

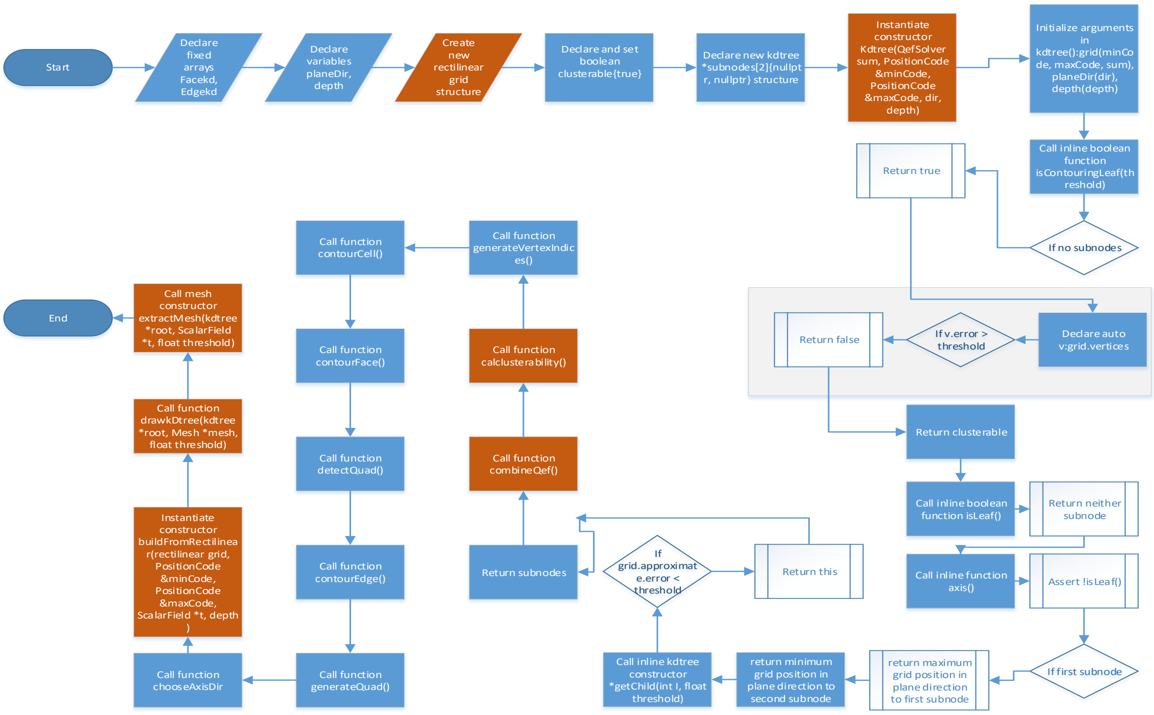

Start
Declare fixed arrays Facekd, Edgekd
Declare variables planeDir, depth
Create new rectilinear grid structure
Declare and set boolean clusterable{true}
Declare new kdtree *subnodes[2]{nullptr, nullptr} structure
Instantiate constructor Kdtree(QefSolver sum, PositionCode &minCode, PositionCode &maxCode, dir, depth)
Initialize arguments in kdtree():grid(minCode, maxCode, sum), planeDir(dir), depth(depth)
Call inline boolean function isContouringLeaf(threshold)
If no subnodes
Return true
Declare auto v:grid.vertices
If v.error > threshold
Return false
Return clusterable
Call inline boolean function isLeaf()
Return neither subnode
Call inline function axis()
Assert !isLeaf()
If first subnode
return maximum grid position in plane direction to first subnode
return minimum grid position in plane direction to second subnode
Call inline kdtree constructor *getChild(int I, float threshold)
If grid.approximate.error < threshold
Return this
Return subnodes
Call function combineQef()
Call function calclusterability()
Call function generateVertexIndices()
Call function contourCell()
Call function contourFace()
Call function detectQuad()
Call function contourEdge()
Call function generateQuad()
Call function chooseAxisDir
Instantiate constructor buildFromRectilinear(rectilinear grid, PositionCode &minCode, PositionCode &maxCode, ScalarField *t, depth)
Call function drawkDtree(kdtree *root, Mesh *mesh, float threshold)
Call mesh constructor extractMesh(kdtree *root, ScalarField *t, float threshold)
End

Figure 3: Flowchart for the implementation of the *k*-d tree algorithm

eral mesh according to the topology preserving scheme. The previous sub-phases 1.1-1.4 supplied the information required for the construction of the embedded polygonal mesh as part of the underlying rectilinear grid. The first step in this sub-phase was to draw an axis aligned bounding box (AABB) enclosing the whole mesh including all vertices and triangles of the polygons. Each vertex contained attributes to be used by the algorithm for positioning and clustering of that particular vertex. The attributes comprised a vertex's index, position and normal in relation with the mesh and the overall rectilinear grid. The algorithm added vertices into the grid according to the associated scalar field representation of the surface. This was performed through QEF based computation of each vertex location with respect to edges of the AABB. The flat normals were generated for each vertex according to its position of each. After that, to ensure that the mesh was topologically correct for watertight and two-manifold and surfaces, the vertices were clustered through the manifold vertex clustering scheme.

3.2 Phase 2: Isosurface extraction through error controlled simplification

The first two sub-phases of isosurface extraction through error controlled simplification were generation of the scalar field of the implicit surface representation (Sub-phase 2.1), and preparation of the computation of topology preserving vertex clustering using QEFs (Sub-phase 2.2). Those standard sub-phases were adopted from Schaefer et al. [40]. The next sub-phase (2.3) involved QEF-controlled isosurface simplification on implicit *k*-d tree or octree based adaptive grids. Our main focus was on utilization of the *k*-d tree for simplification of the rectilinear grid surface representation in order to adaptively contour implicit surfaces of any dimensions. The simplification scheme was adopted from [40] with the exception of a *k*-d tree being utilized instead of an octree. The algorithm performed traversal of the *k*-d tree in a top-down manner. For every leafless *k*-d tree cell, two offspring nodes were considered. Two offspring nodes had faces internal to their associated parent cell along each Euclidean axis. Similar to the previous phase (Sub-phase 1.4), vertices were clustered with consideration to the manifold criterion. The difference was that this time vertices were topologically linked together by their edges twofold to the internal faces of the parent cell. The vertex edges were identified through recursion. In each set, vertices were clustered together through a combination of their QEFs and minimising error functions to evaluate the new vertex's location and its associated error. If the error for evaluating and locating the new vertex was less than the given threshold, then this vertex was marked as collapsible. The subdivision process involved storing a list of faces within the cell and of each individual subdivision step. The faces within the cell were clipped to the partitioning plane in accordance with the manifold criterion. Each of these clipped faces was moved into the associated sub-cell. Information obtained from the clipping process included the orientation of the planes the faces were clipped against. This information was used in detection of vertices associated with the intersection points found on the edges of the grid.

4 Methodology

The *k*-d tree and octree algorithms were implemented on a Windows 10 system with an Intel Core i5 CPU, NVIDIA GeForce MX150 GPU, and 4 GB RAM. The *k*-d tree construction involved recursive subdivision along one dimension, using QEFs for vertex placement and clustering based on manifold criteria. The octree construction used recursive space subdivision into eight octants, applying QEF similarly and ensuring crack-free, manifold surfaces through dual contouring. Both algorithms employed predefined error thresholds to control the level of detail. Performance metrics like build time, extraction time, and triangle count were used for comparative analysis.

To ensure that the evaluation tool would work for implicit surface representations of extracted isosurfaces, implicit models were prepared using an implicit function with a scalar field quantity as an input. An implicit function is a function defined for differentiation of functions containing the variables, which cannot be easily expressed in the form of $y = f(x)$ [51]. An implicit function is a mathematical function taking in more than one related variable at a time.

The implicit function was programmed in C++ with its inputs being the scalar field quantity and the depth of the field. The time (seconds) for extracting an implicit surface representation from the generated implicit model was determined for each of the two dual manifold contouring algorithms, the *k*-d tree- and octree-based, as part of the comparative evaluation process. The extraction of an implicit surface representation was performed in the CPU using the OpenGL API. The rendering of graphics on the screen was carried out in the GPU, also using OpenGL API.

5 Results

5.1 Implicit surface representations for selected objects

Both compared algorithms were run on a variety of implicit models depicted in Figures 4 (a cube), 5 (a sphere), 6 (a cylinder), 7 (a torus), and 8 (a random surface).

The implicit surface representations from the extracted isosurface meshes displayed the solid objects commonly studied or modeled, for example, in computer aided design (CAD) and constructive solid geometry (CSG). The solid volumes were shaded through to reflect the nature of the represented real-life physical objects. The assumption was that the materials depicted in the surface representations were solid in nature, although in reality this might not always be the case. The implicit surface representations considered in this research were computer-generated solid objects extracted from models of implicit functions, and not from volumetric datasets of actual physical objects. The

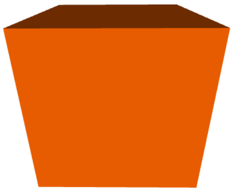


Figure 4: A cube

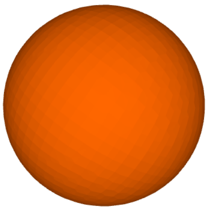


Figure 5: A sphere

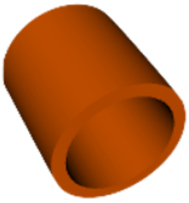


Figure 6: A cylinder

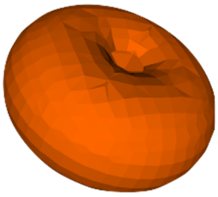


Figure 7: A torus

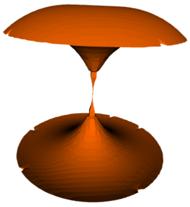


Figure 8: A random surface

manifold dual contouring algorithm was used to extract the embedded mesh and simplify the surface representation of the underlying rectilinear grid for a given error threshold.

Figure 9 shows a wire frame of axis-aligned bounding box (AABB) and its enclosed sphere. This diagram shows how clustered vertices were connected into polygons of the resulting extracted isosurface.

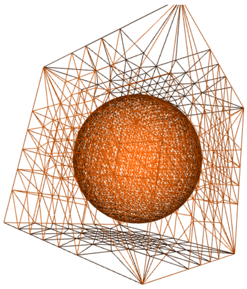


Figure 9: A sphere enclosed inside an AABB

5.2 Results of evaluation of the two algorithms

The two algorithms for isosurface extraction, (k-d tree and octree-based, were evaluated and compared using three parameters, namely, *visualization*, *performance*, and *efficiency*.

In regard to *visualization*, no noticeable difference was observed in the visual graphics extracted by the k-d tree- and octree-based algorithms. Both algorithms utilized the manifold vertex clustering scheme for dual contouring of isosurfaces. The images in Figures 4-9 could have been produced by any of the two algorithms. It could, however, be expected that there might be a difference in the rendering of large models in favor of the octree-based algorithm. An octree-based algorithm would generate more visual details by using a larger number of triangles in its simplification process as compared to the k-d tree algorithm as shown in Tables 3 and 4 (triangle count) for various simplification error thresholds (denoted as Simpl. Err. Threshold). The visual difference also could have been observed if one of the two algorithms was based on an entirely different isosurface extraction approach, e.g., the original marching cubes algorithm.

The performance of each algorithm was measured using two metrics, (1) the time to build the adaptive hierarchy of the data structure (Build Time) and (2) the time taken to extract the implicit surface representations (Extract Time). The values of these metrics for various simplification error thresholds (Simpl. Err. Threshold) are shown in Tables 3 and 4. The comparison between the two algorithms indicated that for building and extraction of an identical implicit surface representation under the same simplification error threshold the k-d three-based algorithm performed better than the octree-based algorithm. For example, in the case of a cylinder model with the simplification error threshold

Table 3: Results for the k-d tree-based manifold dual contouring algorithm

Object Modeled	Simpl. Err. Threshold	Build Time (sec)	Extract Time (sec)	Triangle Count
Cylinder	$1e^{-3}$	0.3349	0.015	2266
	$1e^{-5}$	0.015	0.029	2244
	$1e^{-7}$	0.347	0.015	2280
	$1e^{-9}$	0.351	0.015	2336
Sphere	$1e^{-3}$	0.326	0.036	9190
	$1e^{-5}$	0.333	0.035	9206
	$1e^{-7}$	0.348	0.037	9206
	$1e^{-9}$	0.337	0.036	9206
Torus	$1e^{-3}$	0.095	0.01	2080
	$1e^{-5}$	0.09	0.01	2080
	$1e^{-7}$	0.088	0.01	2080
	$1e^{-9}$	0.089	0.01	2080
Random surface	$1e^{-3}$	0.284	0.033	7626
	$1e^{-5}$	0.284	0.033	7550
	$1e^{-7}$	0.278	0.033	7558
	$1e^{-9}$	0.279	0.033	7558

Table 4: Results for the octree-based manifold dual contouring algorithm

Object Modeled	Simpl. Err. Threshold	Build Time (sec)	Extract Time (sec)	Triangle Count
Cylinder	$1e^{-3}$	1.545	0.028	14303
	$1e^{-5}$	1.588	0.029	14192
	$1e^{-7}$	1.598	0.028	14304
	$1e^{-9}$	1.583	0.028	14400
Sphere	$1e^{-3}$	1.557	0.02	9472
	$1e^{-5}$	1.568	0.02	9464
	$1e^{-7}$	1.558	0.019	9464
	$1e^{-9}$	1.542	0.019	9464
Torus	$1e^{-3}$	1.693	0.007	2080
	$1e^{-5}$	1.712	0.007	2080
	$1e^{-7}$	1.708	0.006	2080
	$1e^{-9}$	1.924	0.007	2080
Random surface	$1e^{-3}$	1.78	0.025	7928
	$1e^{-5}$	1.75	0.02	7928
	$1e^{-7}$	1.754	0.019	7928
	$1e^{-9}$	1.741	0.018	7928

of $1e^{-3}$, a k-d tree-based manifold dual contouring algorithm achieved the 78% reduction in build time and 46% reduction in extract time as compared to the octree-based algorithm. The average reduction of the build and extract times, across all modeled objects and all simplification error thresholds, was 83.4% and 61.2%, respectively, in favor of the k-d tree-based algorithm.

The evaluation of *efficiency* involved determining the numbers of triangles for both algorithms under various simplification error thresholds and for various modeled objects. There is a direct link between efficiency and the triangle counts as the less is the triangle count, the higher is the simplification efficiency. The average improvement in simplification efficiency across all modeled objects and simplification error thresholds for the k-d tree-based algorithm was calculated at 17.7% as compared to the octree-based algorithm. In case of the cylinder model, across all simplification error thresholds the improvement in efficiency was the highest (84.5%) for the k-d tree-based algorithm. However, for the torus model the triangle count across all simplification error thresholds for simplifying the representation of the torus model was the same for both algorithms. This can be explained by the model size because a smaller model with fewer triangles could not be simplified any further by either algorithm. The evaluation results demonstrated that the octree-based algorithm is more effective in preserving the original geometric details of an implicit model. On average, the octree-based algorithm generated a higher number of rectangles, making it the more suitable choice for applications requiring detailed geometric accuracy.

6 Discussion

The comparative analysis conducted in this study revealed several insights into the performance and efficiency of the *k*-d tree- and octree-based manifold dual contouring algorithms in relation to the state of the art. This discussion compares the results of our work with those listed in Table 1, focusing on performance metrics, visual and efficiency outcomes, potential reasons for observed differences, and novel contributions and advancements over the similar existing work.

6.1 Differences in performance metrics

The performance metrics evaluated in this study included build time, extract time, and triangle count. The *k*-d tree-based algorithm demonstrated superior performance in terms of build and extract times compared to the octree-based algorithm. Specifically, the *k*-d tree-based algorithm achieved an average reduction of 83.4% in build time and 61.2% in extract time across all modeled objects and simplification error thresholds. In contrast, previous studies, for example, by Greß and Klein [17], did not provide a direct comparative analysis between *k*-d trees and octrees, thus lacking a performance benchmark for *k*-d tree efficiency.

6.2 Visual and efficiency outcomes

In terms of visual outcomes, no significant differences were observed between the *k*-d tree- and octree-based algorithms when evaluated visually. Both algorithms utilized the manifold vertex clustering scheme, ensuring high-quality isosurface representations. However, the octree-based algorithm excelled in preserving geometric details, producing an average of 20% more triangles than the *k*-d tree-based algorithm. This aligns with findings by Schaefer and Warren [10] and Kim et al. [47], who emphasized the superior visual detail retention of octree-based methods.

Regarding efficiency, the *k*-d tree-based algorithm showed an improvement in simplification efficiency, with an average reduction in triangle count by 17.7% compared to the octree-based algorithm. This efficiency gain is particularly relevant for large implicit models, where reducing geometric complexity is crucial. Previous studies, such as those by Jin et al. [12] and Vignoles et al. [15], highlighted improvements in specific aspects of the marching cubes algorithm but did not achieve the overall efficiency gains demonstrated by the *k*-d tree-based approach in our research.

6.3 Potential reasons for observed differences

The observed differences in performance and efficiency can be attributed to the inherent characteristics of *k*-d trees and octrees. *K*-d trees partition the space adaptively along one dimension at a time, which allows for more flexible and efficient handling of large datasets with varying geometric complexities. This adaptive partitioning reduces the overall geometric complexity more effectively than the simultaneous three-dimensional subdivision used by octrees.

Additionally, the *k*-d tree's ability to minimize build and extract times stems from its binary partitioning scheme, which simplifies the computational process. In contrast, the octree's approach to maintaining high visual detail results in higher triangle counts and longer processing times.

6.4 Study contributions

The contributions of this study to the body of knowledge in computer science, specifically in the areas of computer graphics and geometric modeling, are as follows.

First, unlike previous studies that primarily focused on individual algorithm improvements, this study provides a comprehensive empirical comparison between *k*-d tree- and octree-based manifold dual contouring algorithms. For example, Schaefer et al. [40] focused on octree-based dual contouring methods without juxtaposing them against *k*-d tree methodologies, while others whose research is listed in Table 1 have only improved aspects of individual algorithms.

Second, by evaluating build time, extract time, and triangle count across various simplification error thresholds, this study presents a holistic view of algorithm efficiency. This multi-metric evaluation approach advances the understanding of performance trade-offs in isosurface extraction.

Third, the findings will assist researchers and practitioners in selecting suitable adaptive data structures for different application scenarios. The demonstrated efficiency of the *k*-d tree-based algorithm for large implicit models offers a practical alternative to octree-based methods.

Fourth, the *k*-d tree-based algorithm's superior performance in reducing geometric complexity without compromising visual quality represents a significant advancement in isosurface extraction techniques.

Lastly, the comparative analysis undertaken in this study demonstrated the strengths and weaknesses of *k*-d tree- and octree-based manifold dual contouring algorithms, providing insights for future research and practical implementations in the field of isosurface extraction.

7 Conclusions, limitations and future work

7.1 Conclusions

This paper presented an evaluation of the two manifold dual contouring algorithms using *k*-d trees and octrees across various simplification error thresholds. The first manifold dual contouring octree-based algorithm was adopted from Schaefer et al. [40]. The second manifold dual contouring algorithm was adopted from the same work by Schaefer et al. [40] but a *k*-d tree was employed instead of the octree for simplification as proposed by Greß and Klein in [17]. A rectilinear grid was developed as a prerequisite to capture the initial representation of the underlying mesh, and the QEF was formulated, resulting in the construction of the general mesh according to the topology preserving scheme. The isosurface extraction of implicit models was carried out through the error controlled k-d tree and octree simplification.

The following conclusions were drawn from the evaluation of visualization, performance and efficiency of the compared algorithms. Scaling implicit environments can impact the performance of a manifold dual contouring algorithm due to the processing of excessive geometric details. The k-d tree-based algorithm was found to be more suitable, as this adaptive data structure significantly reduces the amount of geometry in large implicit models. For a scenario requiring more of the geometric detail of the simplified implicit model to be retained, the octree would be considered more suitable because of this data structure's ability to preserve geometric detail as demonstrated by the higher triangle count of the simplification results.

The performance of each algorithm was measured using two metrics, (1) the time taken to build the adaptive hierarchy of the data structure and (2) the time taken to extract the implicit surface representations. The comparison of the two algorithms demonstrated that for building and extraction of a implicit surface representation under the same simplification error threshold the k-d tree-based algorithm performed better than the octree-based algorithm. The average reduction of the build and extract times, across all modeled objects and all simplification error thresholds, was 83.4% and 61.2%, respectively, in favor of the k-d tree-based algorithm. The evaluation of efficiency for the two algorithms involved determining the numbers of triangles for both algorithms under various simplification error thresholds for various modeled objects. The average improvement in simplification efficiency across all modeled objects and simplification error thresholds by the k-d tree-based algorithm was calculated at 17.7% as compared to the octree-based algorithm. However, if geometry had to be preserved as much as possible in order to capture the original geometric details of an implicit model, the octree would be more applicable as, on average, it generated more rectangles.

Research results presented in this paper could be useful for both researchers and practitioners in the area of visualization facing a choice of a suitable adaptive data structure (i.e., a k-d tree or an octree) for better simplification results in implementations of the manifold dual contouring algorithm.

7.2 Study limitations

The study had several limitations that need to be addressed in future research. Firstly, the research scope was restricted to a specific set of implicit models (cube, sphere, cylinder, torus, and random surface). Including additional models with varying complexities could provide a broader evaluation. Secondly, experiments were conducted on a single PC configuration due to resource constraints, which might affect performance results on different hardware setups. Thirdly, the focus was on specific implementations of k-d tree- and octree-based algorithms; however, exploring other variations might offer new insights. Fourthly, as this research used predefined simplification error thresholds, varying these could affect performance metrics and visual quality. Fifthly, further experiments are needed to validate the scalability of both algorithms with larger datasets and more complex implicit models. Lastly, the study did not include visual comparisons of the extracted isosurfaces under different simplification thresholds, which would help illustrate the practical implications of the quantitative results.

7.3 Future work

Future research could include a more diverse set of implicit models to enhance the generalizability of the findings. Evaluating the performance on distributed computing systems or different hardware configurations could provide deeper insights into the scalability and efficiency of the algorithms. Investigating other variations or hybrid approaches of k-d tree and octree-based algorithms might uncover new performance benefits. Implementing adaptive simplification thresholds based on the model's complexity and application requirements could further optimize the algorithms' performance and efficiency. Applying these algorithms to real-world datasets, such as medical imaging or geological surveys, could demonstrate their practical utility and reveal additional areas for improvement. Including visual comparisons of the extracted isosurfaces from both algorithms under different simplification thresholds will help in illustrating the practical implications of the quantitative results presented.

References

[1] C. Hansen, C. Johnson, The visualization handbook, Cambridge, MA: Academic Press, 2004.

[2] W. Wu, L. Ye, Research on real scene robot 3d visualization of historical architectural heritage based on big data objects, Informatica (Slovenia) 48 (2024) 93–102. doi:10.31449/inf.v48i8.5776.

[3] D. Cheng, Animation vr scene stitching modeling based on genetic algorithm, Informatica (Slovenia) 48 (2024) 83–96. doi:10.31449/inf.v48i5.5364.

[4] S. H. Al-Dhaimesh, N. Taib, Review paper: investigation of augmented reality – bim benefits in design process in the aec industry, Informatica (Slovenia) 47 (2023) 111–126. doi:10.31449/inf.v47i5.4671.

[5] H. Fan, B. Goyal, K. Z. Ghafoor, Computer-aided architectural design optimization based on bim tech-

nology, Informatica (Slovenia) 46 (2022) 323–332. doi:10.31449/inf.v46i3.3935.

[6] I. Bankman, Handbook of medical image processing and analysis, Cambridge, MA: Academic Press, 2009.

[7] J. O'Brien, G. Turk, Modelling with implicit surfaces that interpolate, ACM Transactions on Graphics 21 (2002) 855–873. doi:10.1145/571647.571650.

[8] M. Edmunds, R. Laramee, G. Chen, N. Max, E. Zhang, C. Ware, Surface-based flow visualization, Computers Graphics 36 (2012) 974–990. doi:10.1016/j.cag.2012.07.006.

[9] W. Lorensen, H. Cline, Marching cubes: A high resolution 3d surface construction algorithm, ACM SIGGRAPH Computer Graphics 21 (1987) 163–169.

[10] S. Schaefer, J. Warren, Dual marching cubes: Primal contouring of dual grids, in: Proceedings of the Pacific Conference on Computer Graphics and Applications, 2004, pp. 70–76. doi:10.1109/pccga.2004.1348336.

[11] S.-W. Lee, H.-Y. Jung, R. Prost, A. Senot, Regularized marching cubes mesh, in: Proceedings of the IEEE International Conference on Image Processing, Vol. 3, 2005, pp. 788–791. doi:10.1109/icip.2005.1530510.

[12] J. Jin, Q. Wang, Y. Shen, J. Hao, An improved marching cubes method for surface reconstruction of volume data, in: Proceedings of the World Congress on Intelligent Control and Automation, Vol. 2, 2006, pp. 10454–10457. doi:10.1109/wcica.2006.1714052.

[13] R. Strand, P. Stelldinger, Topology preserving marching cubes-like algorithms on the face-centered cubic grid, in: Proceedings of the 14th International Conference on Image Analysis and Processing, 2007, pp. 781–788. doi:10.1109/iciap.2007.4362871.

[14] Z. Xu, C. Xiao, X. Xu, An improved marching cubes algorithm based on edge contraction, in: Proceedings of the International Conference on Signal Processing Proceedings, 2010. doi:10.1109/icosp.2010.5655719.

[15] G. Vignoles, M. Donias, C. Mulat, C. Germain, J.-F. Delesse, Simplified marching cubes: an efficient discretization scheme for simulations of deposition/ablation in complex media, Computational Materials Science 50 (2011) 893–902. doi:10.1016/j.commatsci.2010.10.027.

[16] Q. Du, J. Zhao, L. Shi, L. Wang, Efficient improved marching cubes algorithm, in: Proceedings of the 2nd International Conference on Computer Science and Network Technology, 2012, pp. 416–419. doi:10.1109/iccsnt.2012.6525967.

[17] A. Greß, R. Klein, Efficient representation and extraction of 2-manifold isosurfaces using kd-trees, in: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, 2003, pp. 370–397. doi:10.1109/pccga.2003.1238278.

[18] J. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM 18 (1975) 509–517. doi:10.1145/361002.361007.

[19] D. Meagher, Octree generation, analysis and manipulation, Ph.D. thesis, Rensselaar Polytechnic Institute, Troy New York (1982).

[20] T. Ju, F. Losasso, S. Schaefer, J. Warren, Dual contouring of hermite data, ACM Transactions on Graphics 21 (2002) 339–346. doi:10.1145/566654.566586.

[21] J.-D. Boissonnat, S. Oudot, Provably good sampling and meshing of surfaces, Graphical Models 67 (2005) 405–451. doi:10.1016/j.gmod.2005.01.004.

[22] M. Garland, P. Heckbert, Surface simplification using quadric error metrics, in: Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, 1997, p. 209–216. doi:10.1145/258734.258849.

[23] Y. Livnat, C. Hansen, C. Johnson, Isosurface extraction for large-scale data sets, in: Proceedings of the Conference on Data Visualization: The State of the Art, 2003, pp. 77–94. doi:10.1007/978-1-4615-1177-9_6.

[24] T. Newman, H. Yi, A survey of the marching cubes algorithm, Computers Graphics 30 (2006) 854–879. doi:10.1016/j.cag.2006.07.021.

[25] D. Rajon, W. Bolch, Marching cube algorithm: review and trilinear interpolation adaptation for image-based dosimetric models, Computerized Medical Imaging and Graphics 27 (2003) 411–435. doi:10.1016/s0895-6111(03)00032-6.

[26] C. Maple, Geometric design and space planning using the marching squares, in: Proceedings of the International Conference on Geometric Modeling and Graphics, 2003, pp. 90–95. doi:10.1109/gmag.2003.1219671.

[27] C. Andujar, P. Brunet, A. Chica Calaf, J. Rossignac, A. Vinacua, Optimizing the topological and combinatorial complexity of isosurfaces, Computer-Aided Design 37 (2005) 847–857. doi:10.1016/j.cad.2004.09.013.

[28] X. Renbo, L. Weijun, Y. Wang, A robust and topological correct marching cube algorithm without look-up

table, in: Proceedings of the 5th International Conference on Computer and Information Technology, 2005, pp. 565–569. doi:10.1109/cit.2005.44.

[29] N. Chrisochoides, D. Nave, Simultaneous mesh generation and partitioning for delaunay meshes, Mathematics and Computers in Simulation 54 (1999) 321–339. doi:10.1016/s0378-4754(00)00192-0.

[30] S. H. Cui, J. Liu, Simplified patterns for extracting the isosurfaces of solid objects, Image and Vision Computing 26 (2008) 339–346. doi:10.1016/j.imavis.2007.02.003.

[31] T. Etiene, C. Scheidegger, L. Nonato, R. Kirby, C. Silva, Verifiable visualization for isosurface extraction, IEEE Transactions on Visualization and Computer Graphics 15 (2009) 1227–1234. doi:10.1109/tvcg.2009.194.

[32] Q. Xiaoqing, W. Davis, An extended cuberille model for identification and display of 3d objects from 3d gray value data, in: Proceedings of the Conference on Graphics Interface, 1992, pp. 70–77.

[33] L. Custodio, T. Etiene, S. Pesco, C. Silva, Practical considerations on marching cubes 33 topological correctness, Computers Graphics 37 (2013) 840–850. doi:10.1016/j.cag.2013.04.004.

[34] E. Tcherniaev, Marching cubes 33: Construction of topologically correct isosurfaces, Tech. rep., Institute for High Energy Physics (1996).

[35] T. Lewiner, H. Lopes, A. Vieira, G. Tavares, Efficient implementation of marching cubes' cases with topological guarantees, Journal of Graphics Tools 8 (2012) 1–15. doi:10.1080/10867651.2003.10487582.

[36] J. Chen, X. Jin, Gpu-based polygonization and optimization for implicit surfaces, The Visual Computer 31 (2014) 119–130. doi:10.1007/s00371-014-0924-7.

[37] T. Athawale, E. Sakhaee, E. Alireza, Isosurface visualization of data with nonparametric models for uncertainty, IEEE Transactions on Visualization and Computer Graphics 22 (2015) 777–786. doi:10.1109/tvcg.2015.2467958.

[38] Y. Ronghuan, X. Wei, W. Lingda, H. Hongxing, Research on multi-resolution isosurface extraction method for 3d scalar field, in: Proceedings of the 2nd IEEE International Conference on Data Science in Cyberspace, 2017, pp. 359–362. doi:10.1109/DSC.2017.79.

[39] N. Zhang, W. Hong, A. Kaufman, Dual contouring with topology-preserving simplification using enhanced cell representation, in: Proceedings of the IEEE Visualization Conference, 2004, pp. 505–512. doi:10.1109/visual.2004.27.

[40] S. Schaefer, T. Ju, J. Warren, Manifold dual contouring, IEEE Transactions on Visualization and Computer Graphics 13 (2007) 610–619. doi:10.1109/tvcg.2007.1012.

[41] Y. Zhang, J. Qian, Dual contouring for domains with topology ambiguity, Computer Methods in Applied Mechanics and Engineering s 217–220 (2012) 34–45. doi:10.1016/j.cma.2012.01.004.

[42] X. Liang, Y. Zhang, An octree-based dual contouring method for triangular and tetrahedral mesh generation with guaranteed angle range, Engineering with Computers 30 (2014) 211–222. doi:10.1007/s00366-013-0328-8.

[43] A. Peixoto, C. Moura, Topology preserving algorithms for implicit surfaces simplifying and sewing, in: Proceedings of the International Conference on Computational Science and Its Applications, 2014, pp. 352–367. doi:10.1007/978-3-319-09129-7_27.

[44] T. Rashid, S. Sharmin, M. Audette, Watertight and 2-manifold surface meshes using dual contouring with tetrahedral decomposition of grid cubes, Procedia Engineering 163 (2016) 136–148. doi:10.1016/j.proeng.2016.11.037.

[45] G. Varadhan, S. Krishnan, Y. Kim, D. Manocha, Feature-sensitive subdivision and isosurface reconstruction, in: Proceedings of the IEEE Visualization Conference, 2003, pp. 99–106. doi:10.1109/visual.2003.1250360.

[46] G. Nielson, Dual marching cubes, in: Proceedings of the IEEE Visualization Conference, 2004, pp. 489–496. doi:10.1109/visual.2004.28.

[47] S. Kim, Y. Ohtake, Y. Nagai, H. Suzuki, A novel interpolation scheme for dual marching cubes on octree volume fraction data, Computers and Graphics 66 (2017) 169–178. doi:10.1016/j.cag.2017.05.021.

[48] R. Grosso, D. Zint, A parallel dual marching cubes approach to quad only surface reconstruction, The Visual Computer 38 (2022) 1301–1316. doi:10.1007/s00371-021-02139-w.

[49] Y. He, M. Mirzargar, S. Hudson, M. Kirby, R. Whitaker, An uncertainty visualization technique using possibility theory: Possibilistic marching cubes, International Journal for Uncertainty Quantification 5 (2015) 433–451. doi:10.1615/int.j.uncertaintyquantification.2015013730.

[50] D. El-Rushaidat, R. Yeh, X. Tricoche, Accurate parallel reconstruction of unstructured datasets on rectilinear grids, Journal of Visualization 24 (2021) 787–806. doi:10.1007/s12650-020-00740-0.

[51] J. Stewart, Calculus early transcendentals, Cengage Learning, 2015.

A Framework for Malicious Domain Names Detection Using Feature Selection and Majority Voting Approach

Dharmaraj R. Patil
Department of Computer Engineering, R.C.Patel Institute of Technology, Shirpur, India
E-mail: dharmaraj.patil@rcpit.ac.in

Keywords: malicious domain names, cybersecurity, feature selection, majority voting, machine learning

Received: March 2, 2024

As cyber attacks become more sophisticated, identifying and mitigating bad domain names has become critical to assuring the security of online environments. This paper presents a framework for detecting malicious domain names using a feature selection strategy and a majority vote method. The suggested methodology begins with the extraction of important features from domain names and their related characteristics, followed by a rigorous feature selection procedure to determine the most discriminating attributes. To accomplish feature selection, a variety of feature selection techniques are used, including chi-square statistics, information gain, gain ratio, and correlation-based feature selection, to analyse the value of each characteristic in distinguishing benign and malicious domain names. In addition, a majority voting strategy is utilised to improve the detection system's overall accuracy and reliability by combining the predictions of different classifiers such as AdaBoost, logistic regression, k-nearest neighbours, naive bayes, and multilayer perceptron. The ensemble of classifiers is trained on the ideal features, yielding a complete and robust model capable of accurately recognising malicious domain names while minimising false positives. The proposed approach is evaluated against real-world examples of harmful domain names. The suggested framework employing Chi-square feature selection and majority voting detects malicious domain names with an accuracy of 99.44%, precision of 99.44%, recall of 99.44%, and f-measure of 99.44%. The use of feature selection and a majority voting technique improves the system's adaptability and resilience in the face emerging cyber threats.

Povzetek: Raziskava predstavlja okvir za zaznavanje zlonamernih domen, ki uporablja statistično selekcijo značilnosti in pristop večinske izbire ter strojno učenje.

1 Introduction

In an era characterized by the pervasive influence of the digital landscape, the escalating sophistication of cyber threats poses a significant challenge to the security of online environments. Malicious actors exploit various avenues to compromise the integrity and confidentiality of information, and one such vector is the utilization of malicious domain names. These deceptive entities serve as a pivotal component in orchestrating cyber-attacks, making their timely detection an imperative aspect of cybersecurity. As per the Cybercrime Information Center, there has been a substantial rise in the utilization of domain names within malware URLs. A study conducted by Interisle revealed a significant surge, indicating a 121% increase in the prevalence of domain names in the fourth quarter of 2022 [1].

Additionally, as outlined in the CSC Domain Security 2023 report [2]:

- 43% of .AI domains are registered by entities not associated with the original owners.
- 21% of DNS records from subdomains direct to unresolvable content, exposing organizations to the risk of

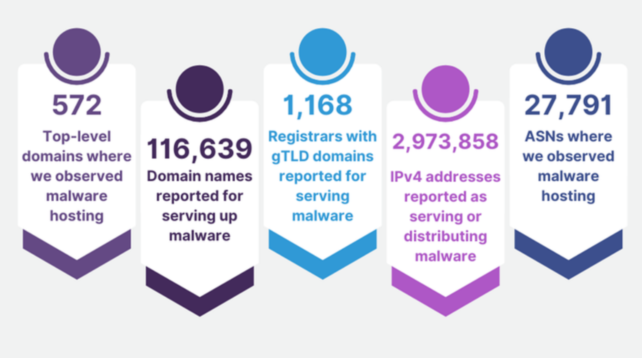


Figure 1: Interisle malicious domain names statistics 4Q 2022. *Source:* `https://www.cybercrimeinfocenter.org/malware-andscape-2023`

subdomain hijacking.

- 79% of registered domains bearing resemblance to global 2000 brands (homoglyphs) are under the ownership of third parties.

- 46% of enterprises employing enterprise-class registrars have also implemented registry locks.

- A noteworthy 112 companies exhibit a domain security score of 0%.

- Domain-based message authentication, reporting, and conformance have grown by 6%, the greatest increase in the past four years.

This study focuses on the design of a detection system for fraudulent domain names that takes advantage of the synergy between optimal feature selection and a majority voting technique. The rising complexity of cyber threats needs new and adaptable methods for identifying and mitigating possible dangers. By tackling the issues of detecting bad domain names, this study helps to improve the overall resilience of online systems to online attacks. To build a strong framework, the suggested methodology combines machine learning concepts, feature selection, and ensemble classifiers. It is the goal of a careful feature selection procedure to uncover the most discriminatory qualities that discriminate between benign and malicious domain names. This ideal feature subset not only improves the detection system's performance, but it also helps it adapt to changing threat landscapes. Furthermore, using a majority voting mechanism strengthens the detection system's dependability. By aggregating several classifier predictions, the model improves its ability to make accurate decisions, reduce false positives, and improve overall system performance. The ensemble of classifiers, trained on the optimal characteristics chosen, forms a cohesive defence system against the ever-changing nature of cyber attacks. The suggested framework, which uses Chi-square feature selection and majority voting, detects malicious domain names with an accuracy of 99.44%, precision of 99.44%, recall of 99.44%, and f-measure of 99.44%. The main outcomes of this study are as follows:

- This study introduces a strong methodology for detecting fraudulent domain names, providing a solution to the expanding issues provided by cyber attacks in online environments.

- The proposed system includes an optimal feature selection strategy based on modern machine learning algorithms and statistical techniques. Using Chi-square feature selection and majority voting, the framework achieves exceptional performance, with accuracy, precision, recall, and f-measure all registering at 99.44%.

- The proposed framework's efficacy is carefully evaluated using a variety of datasets, including real-world examples of malicious domain names. This extensive test assures the detection system's practical usefulness and resilience in a variety of tough settings.

The remaining sections of this work are organised as follows. Section 2 presents the motivation for the malicious domain names detection. Section 3 offered a brief summary of pertinent past studies. Section 4 described the methodology that included feature selection techniques and supervised batch machine learning algorithms. Section 5 has a full account of the findings. Section 6 presents the conclusive findings.

2 Motivation

The identification and mitigation of malicious domain names are crucial cybersecurity efforts. With an increased reliance on the internet for different parts of everyday life, ranging from communication and commerce to critical services and infrastructure, the threat presented by hostile actors using domain names for evil purposes has grown dramatically. Malicious domain names serve as a conduit for a variety of cyber dangers, such as phishing attempts, malware distribution, and botnet command and control infrastructure. As a result, comprehensive and effective procedures for detecting and combating these threats are urgently required to protect persons, organisations, and the integrity of the digital ecosystem.

Traditional techniques to malicious domain name identification frequently depend on human analysis or static rule-based systems, both of which are fundamentally constrained in their capacity to react to the dynamic and complex nature of current cyber threats. Furthermore, the sheer amount of domain names produced everyday renders human examination impracticable and resource-intensive. As a result, there is a rising need to create automated, data-driven approaches for detecting malicious domain names while minimising false positives and reacting to changing threat environments.

The proposed approach is motivated by the need to solve these difficulties by using the capabilities of machine learning, feature selection approaches, and ensemble methods to improve the accuracy and scalability of malicious domain name identification. Our approach seeks to distinguish benign and malicious domain names with high precision and efficiency by utilising powerful algorithms and relevant data retrieved from domain name properties, DNS traffic, and contextual information. Furthermore, the use of a majority voting strategy allows for the combining of predictions from several classifiers, enhancing robustness and resilience against adversarial evasion tactics. This ensemble technique not only improves detection accuracy but also provides a mechanism for responding to emergent threats in real time.

3 Related work

The field of malicious domain name identification has received a lot of interest in recent years, with academics and practitioners working to build efficient solutions to tackle emerging cyber threats. This section provides a quick summary of important past research, emphasising major contributions and approaches used in the topic.

Hong Zhao et al. devised an N-Gram-based technique for identifying fraudulent domain names. The method makes use of Alexa's top 100,000 domain names from 2013, with N-Gram segmentation applied to each domain name except the top-level domain. Substrings of varying lengths (3 to 7) are then formed based on the domain levels. A substring set is defined, and the weight of each substring is calculated by its frequency in the set. To detect malicious assaults, the N-Gram technique is used to segment a given domain name, and its reputation value is calculated using the weights of its substrings. Thresholding determines the domain's harmful character. Experiments using Alexa 2017 and the Malware domain list showed that the algorithm was successful, with an accuracy rate of 94.04%, a false negative rate of 7.42%, and a false positive rate of 6.14%. Notably, the suggested technique demonstrates reduced temporal complexity than existing current fraudulent domain name identification algorithms [3].

Ali Soleymani et al. investigated the DNS network protocol using machine learning methods and text mining techniques, with a special focus on botnet detection. The investigation involves extracting and labelling domain name datasets that contained both healthy and contaminated Domain Generation Algorithm botnet data. To support this research, text-mining-based data preparation methods such as n-gram analysis and Principal Component research were used. The use of PCA includes the extraction of statistical characteristics to improve model performance. The suggested model's performance was evaluated using a variety of machine learning classifiers, including decision tree, support vector machine, random forest, and logistic regression. Experimental results reveal that the random forest algorithm demonstrates significant efficacy in botnet identification, demonstrating the greatest accuracy across classifiers [4].

Luhui Yang et al. presented a domain name syntax model to improve the identification accuracy of algorithmically created domain names. This approach examines many aspects inside domain names and their syntactic links. An adaptive embedding approach is presented to make domain name element processing more efficient. The authors also provide a parallel convolutional model with a feature selection module and an upgraded dynamic loss function based on curriculum learning. This approach performs well in recognising multi-element fraudulent domain names. In a series of tests, the suggested model is tested against five current methods. The findings show that the suggested model outperforms the comparison methods in recognising multiple-element malicious domain names [5].

Shaojie Chen et al. implemented a meaningful word segmentation technique to define the structure of dictionary-based Algorithmically Generated Domains (AGDs). In this study, they propose using standard deviation to improve the assessment of word distribution properties. In addition, an 11-dimensional statistical feature set is created to enhance the findings of word segmentation. The authors then improve detection performance for both character- and dictionary-based AGDs by using 3-gram and 1-gram sequence characteristics. The final phase is feature fusion, which combines the four types of features stated above, resulting in an end-to-end detection approach for both character-based and dictionary-based AGD. The proposed technique achieved an overall accuracy of 97.24% based on experimental assessments. Specifically, it beat previous approaches in terms of accuracy and F1 values on both dictionary-based and character-based AGD datasets [6].

Atif Ali Wagan et al. have developed a new unified learning technique that uses both numerical and linguistic aspects of domain names to determine if a particular domain name combination is harmful. The trials were conducted on a benchmark dataset of 90,000 domain names. The experimental findings show that the suggested strategy outperforms six comparative approaches in terms of accuracy, precision, recall, and F1-Score. This study makes a significant addition to domain name categorization by presenting a unified learning framework that outperforms previous algorithms across several performance measures [7].

Leyla Bilge et al. introduced EXPOSURE, a system aimed to detect rogue domains using large-scale, passive DNS analysis techniques. The method is based on 15 unique characteristics retrieved from DNS data, which allow for the characterisation of various DNS name attributes and querying patterns. The approach's scalability was evaluated using a large real-world dataset including 100 billion DNS queries. A two-week practical implementation within an ISP also confirmed the system's capacity to automatically detect previously undiscovered harmful domains used in different malicious activities, such as botnet command and control, spamming, and phishing. This work demonstrates the scalability and efficiency of the EXPOSURE system in detecting and mitigating harmful domains across varied malicious actions [8].

Zhaoshan Fan et al. have presented PUMD, a unique framework for detecting malicious domains. To solve the issue of insufficient label information, this system uses a novel Positive and Unlabeled (PU) learning method. To address class imbalance, a customised sample weight technique is used, and evidence features are efficiently built using resource overlapping to reduce the intra-class distance of malicious samples. Furthermore, a feature selection technique based on permutation significance and binning is given to discover the most informative detection features. Experiments on the open-source actual DNS traffic dataset given by QI-ANXIN Technology Group evaluate the PUMD framework's usefulness in collecting probable

C&C domains for harmful activity. These trials show that PUMD consistently provides higher detection performance across various label frequencies and class imbalance ratios [9].

Luhui Yang et al. investigated inter-word and inter-domain correlations using semantic analysis methods, with a special emphasis on word embedding and part-of-speech considerations. The researchers provide a detection framework customised for word-based Domain Generation Algorithms, which incorporates word frequency distribution and part-of-speech into feature set creation. To assess the suggested technique, the ensemble classifier, which includes Naive Bayes, Extra-Trees, and Logistic Regression, is used with both malicious and valid domain samples retrieved from publicly available datasets. When compared to three state-of-the-art DGA detection techniques, the experimental results show that the suggested scheme is much more accurate in identifying word-based DGAs [10].

Yong Shi et al. presented a machine learning framework for detecting malware domain names that makes use of the Extreme Learning Machine (ELM). Their technique is built on ELM, which is well-known for its high accuracy and quick learning capabilities. The researchers used ELM to categorise domain names based on information derived from several sources. Experiments reveal that the suggested detection approach has a high detection rate and accuracy, exceeding 95%. Furthermore, comparison trials demonstrate the quick learning pace of their ELM-based technique. As a result, the researchers say that their approach using ELM is not only successful but also efficient in identifying fraudulent sites [11].

Yu Fu et al. developed two Domain Generation Algorithms that use Hidden Markov Models and Probabilistic Context-Free Grammars , respectively. Their experimental results indicate that traditional DGA detection metrics such as KL, JI, and ED, as well as detection tools such as BotDigger and Pleiades, have difficulties in recognising domain names formed using these methods. To overcome this, the researchers use game theory to optimise methods for botmasters and security professionals. The findings show that, for optimal DGA detection, security personnel should prioritise ED detection with a probability of 0.78 and JI detection with a probability of 0.22.Botmasters should choose the HMM-based DGA with a probability of 0.67 and the PCFG-based DGA with a probability of 0.33, respectively. This study improves DGA detection tactics by using game theory ideas to optimise detection methodologies for both security staff and botmasters [12].

Xiaochun Yun et al. proposed Khaos, a unique Domain Generation Algorithm that uses neural language models and the Wasserstein Generative Adversarial Network to provide strong anti-detection capabilities. The researchers' major finding concerns the construction of genuine domain names, which frequently consist of legible syllables and acronyms. Using this knowledge, Khaos organises syllables and acronyms in neural language models to simulate genuine domain names. Using the Khaos framework, the researchers first determine the most prevalent n-grams in actual domain names. They then tokenize these domain names into n-grams and create new domain names by learning n-gram arrangements from existing domain names. Experimental assessments were carried out employing a variety of cutting-edge DGA detection methodologies, including statistics-based, distribution-based, LSTM-based, and graph-based methods. The experimental results show that Khaos poses significant challenges for existing detection approaches, with an average distance of 0.64 for detecting Khaos using the distribution-based detection approach, AUCs of 0.76 and 0.57 for the statistics-based and LSTM-based detection approaches, and a precision of 0.68 for Khaos using the graph-based detection approach. This emphasises Khaos' greater anti-detection performance compared to state-of-the-art DGAs [13].

Luhui Yang et al. examined the character-level properties of Sub-domain Domain Generation Algorithm) domain names and suggested a new detection framework called Heterogeneous Deep Neural Network. The HDNN framework employs a novel Improved Parallel Convolutional Neural Network architecture that incorporates multi-sized convolution kernels to extract multi-scale local information from domain names. The system also includes a revolutionary Self-Attention-based Bidirectional Long Short-Term Memory architecture, which uses an attention mechanism to extract bidirectional global information from domain names. Furthermore, the researchers adopt a focused loss function to overcome the sample quantity imbalance during the training period. Benchmark studies were carried out on a database of benign domain names, real-world DGAs, and SDGAs. Six popular deep-learning-based DGA detection techniques were compared. The findings show that the suggested technique outperforms state-of-the-art detection for SDGAs while also excelling in binary and multiclass classification for standard DGAs. This research adds to the improvement of DGA detection by presenting a unique framework suited exclusively for SDGAs, demonstrating higher performance in comparison to previous approaches [14].

Congyuan Xu et al. devised a unique strategy by integrating n-gram analysis with deep convolutional neural networks, resulting in the establishment of a novel n-gram combined character-based domain classification model. This model is end-to-end, therefore no manually extracted characteristics or domain name system (DNS) contextual information are required. It just requires the domain name as input, allowing for the automated assessment of the chance that the domain name was generated by Domain Generation Algorithms (DGAs). Experiments on real-world data show that the proposed technique effectively detects domain names created by DGAs, with an average detection rate of 98.69% and an average F-measure of 0.9829. The suggested method outperforms state-of-the-art algorithms in recognising pronounceable and wordlist-based DGA domain names, with a detection rate of over 93.89%. Thus, the suggested detection approach is resilient and versatile in recognising various forms of domain names created by DGAs. This study marks a substantial leap in DGA detection strategies [15].

R. Vinayakumar et al. obtained DNS logs only from client PCs on a local area network (LAN) and stored them on a central server. The researchers recommended using deep learning to determine if a domain name is benign or harmful. They performed a comparison analysis to assess the performance of several deep learning approaches, such as recurrent neural networks (RNN), long short-term memory (LSTM), and classic machine learning classifiers. The study found that deep learning-based techniques outperformed standard machine learning classifiers. This advantage is due to deep learning algorithms' inherent ability to implicitly collect significant characteristics. Notably, LSTM outperformed other deep learning algorithms in terms of malicious detection rate across all studies. This study highlights the effectiveness of deep learning approaches in DNS log analysis for detecting malicious domains within LAN settings [16].

Luhui Yang et al. constructed a semantic element representation model for domain names using a probabilistic context-free grammar model and a collection of semantic components linked with domain names. The model starts by analysing and categorising the domain names' basic parts. It then proposes a syntax tree analysis approach for establishing semantic links between these components, allowing for the efficient encoding of many items inside domain names. This methodology classifies harmful domain names into four types: random character-based, word-based, predicted character-based, and multi-element hybrid. The researchers discovered considerable differences between harmful and normal domain names, as well as between other types of malicious domain names, using tests aimed to analyse anomalies and concealing patterns in domain names. The comparative experimental findings demonstrate the efficiency of the suggested methodology in improving the accuracy of malicious domain name detection. This study adds to the improvement of domain name analysis approaches by presenting a semantic element representation model that efficiently distinguishes between malicious and lawful domain names based on their structural properties [17].

4 Methodology

Figure 2 depicts the suggested architecture for detecting malicious domain names. The framework consists of many steps, including data collection and preprocessing, feature selection, model creation, and the use of a majority voting technique. Each phase substantially improves the overall effectiveness of the malicious domain name detection system.

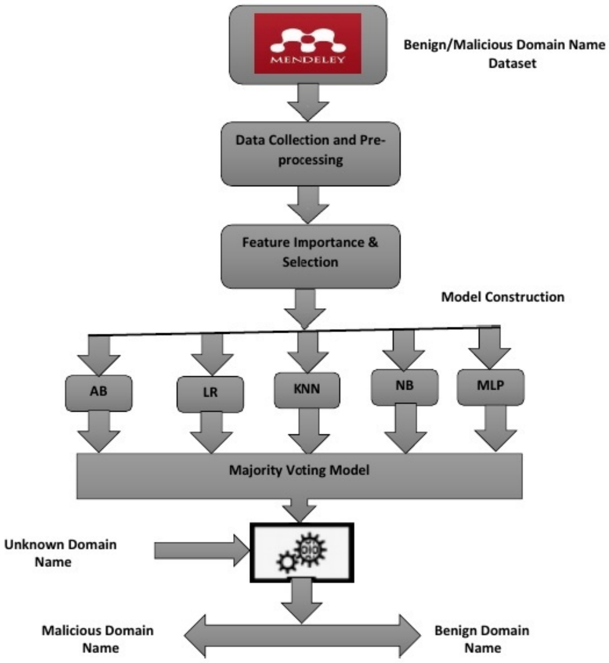


Figure 2: Proposed Framework for Malicious Domain Name Detection System

4.1 Data collection and preprocessing

This study uses the Mendeley Dataset, which includes both benign and malicious domains retrieved from DNS logs [18]. This dataset is especially designed for supervised machine learning research to differentiate between harmful and non-malicious domain names. It was rigorously curated by combining publicly accessible DNS logs from both sorts of domain names. Each domain name is used as an input in the dataset, resulting in 34 characteristics. Domain name properties such as entropy, the occurrence of unique characters, and domain name length are examples of features that are directly extracted. Furthermore, supplemental details such as domain creation date, related IP address, open ports, and geolocation were obtained by data enrichment methods that used Open Source Intelligence methodologies. This collection of 90,000 domain names is rigorously balanced, providing an equal mix of 50% non-malicious and 50% malicious domains.

Five features are removed from the training and testing datasets during the preparation stage: DNSRecordType, CountryCode, RegisteredCountry, RegisteredOrg, and TLD. As a result, the final experimental dataset has 29 characteristics that will be used for model training and testing.

4.2 Feature selection techniques used for malicious domain names detection

Feature selection strategies are used to determine the most discriminating characteristics. To determine the relevance of features to the classification job, many approaches were used, including Chi-square, Gain Ratio, Information Gain, and Correlation-based feature selection. These strategies were used to rank the significance of traits based on their discriminating potential. As a result, we chose a collection of attributes that offer the most important contributions to detecting fraudulent domain names while reducing repetition. We use the Weka implementation of the aforementioned feature selection approaches [19].

4.2.1 Chi-square (χ^2) feature selection technique

Chi-square (χ^2) feature selection is an effective strategy for selecting the most informative features in classification problems. It computes the chi-square statistic to assess the relationship between each characteristic and the target variable. This statistic measures the amount of correlation between categorical variables, allowing us to determine which traits are most relevant for predicting the target variable [20]. The chi-square statistic is computed using the formula below.

$$\chi^2 = \sum \frac{(O - E)^2}{E} \tag{1}$$

Where:

- χ^2 is the chi-square test statistic.
- O is the observed frequency (actual count) of each category or combination of categories in the dataset.
- E is the anticipated frequency (theoretical count) of each category or combination of categories, assuming feature and target variable independence.

Once computed, the chi-square statistic is used to determine the degree of independence between the feature and the target variable. A greater chi-square value implies a stronger link between the characteristic and the goal, whereas lower values imply a weaker connection. In the context of feature selection, characteristics with higher chi-square values are judged more meaningful for predicting the target variable. Table 1 shows the results of the Chi-square (χ^2) Feature Selection, which highlights the most important traits. The Chi-square Feature Selection approach has a threshold of 26067.5297. According to Table 1, a thorough selection procedure resulted in the identification of the 12 most relevant attributes from a total of 28 in the Mendeley DNS dataset.

4.2.2 Gain ratio feature selection technique

Gain Ratio feature selection stands out as a useful strategy for prioritising characteristics that contribute the most to classification problems. This approach evaluates each feature's importance by computing the gain ratio, which considers its inherent qualities and capacity to minimise uncertainty in predicting the target variable [21]. It is computed using the formula below.

$$IGR(X, Y) = \frac{IG(X, Y)}{H(X)} \tag{2}$$

Where:

- $IG(X, Y)$ is the Information Gain between feature X and target variable Y.
- $H(X)$ is the entropy of feature X.

The entropy $H(X)$ for a discrete feature X is calculated as:

$$H(X) = -\sum_i P(x_i) \cdot \log_2(P(x_i)) \tag{3}$$

Where:

- $P(x\_i)$ represents the likelihood of a feature X having a value x_i.

In the area of Information Gain Ratio, the intrinsic information of characteristics is taken into account, yielding a normalised measure. This normalisation is useful when dealing with characteristics that have varied sizes or numbers of possible values. The Information Gain Ratio balances any bias towards features with a large number of different values, which may have a high Information Gain owing to variability. Table 2 illustrates how the information gain ratio is used to identify the most relevant attributes. The information gain ratio feature selection approach, which has a threshold of 0.105, helps with this procedure. A rigorous selection procedure resulted in the identification of the 12 most significant characteristics from the Mendeley DNS Dataset displayed in Table 2.

4.2.3 Information gain feature selection technique

Among the several feature selection approaches, Information Gain feature selection is a crucial way for determining the most significant qualities in classification problems. It uses information theory ideas to measure how much information is acquired about the target variable by incorporating a certain feature into the model. This strategy prioritises characteristics that reduce uncertainty about the target variable, increasing the model's predictive power [22]. The Information Gain (IG) for a given characteristic X with regard to a target variable Y is commonly computed using the following formula:

$$IG(X, Y) = H(Y) - H(Y|X) \tag{4}$$

Where:

- $H(Y)$ represents the entropy of the target variable Y without regard for any special attribute.
- $H(Y|X)$ represents the conditional entropy of Y in the presence of the feature X.

Feature No.	Chi-square Statistic	Feature Name
f1	90000	Domain
f2	73423.4809	NumericSequence
f3	72239.5233	NumericRatio
f4	70611.0287	IP
f5	59096.0642	StrangeCharacters
f6	59074.1194	ASN
f7	55600.5647	ConsoantRatio
f8	51850.5355	DomainLength
f9	45977.5991	VowelRatio
f10	43233.5773	SubdomainNumber
f11	27028.8843	HasSPFInfo
f12	26067.5297	TXTDnsResponse

Table 1: Feature selection using Chi-square Statistic on Mendeley DNS Dataset

Feature No.	Gain Ratio	Feature Name
f1	0.437532	NumericSequence
f2	0.393146	NumericRatio
f3	0.364636	SubdomainNumber
f4	0.257425	ConsoantRatio
f5	0.22911	HasSPFInfo
f6	0.220428	TXTDnsResponse
f7	0.218858	StrangeCharacters
f8	0.213575	VowelRatio
f9	0.162423	CreationDate
f10	0.148529	ASN
f11	0.129045	IP
f12	0.105742	DomainLength

Table 2: Feature selection using Gain Ratio method on Mendeley DNS Dataset

The entropy $H(Y)$ is calculated as:

$$H(Y) = -\sum_{i} P(y_i) \cdot \log_2(P(y_i)) \quad (5)$$

Where:

- $P(y_i)$ represents the probability of class y_i in the target variable Y.

The conditional entropy $H(Y|X)$ is calculated as:

$$H(Y|X) = \sum_{j} P(x_j) \cdot H(Y|X = x_j) \quad (6)$$

Where:

- $P(x_j)$ is the probability of the occurrence of feature X having value x_j.
- $H(Y|X = x_j)$ is the target variable's entropy when the feature X has the value x_j.

In practice, Information Gain is used as a metric to analyse the efficacy of a certain characteristic in distinguishing between distinct classes within the target variable. Higher information gain values indicate more discriminating power. As a result, the feature with the largest Information Gain is judged the most useful and is usually prioritised for further analysis or model training. Table 3 illustrates how information gathering was used to pick the most relevant attributes. The selection of information gain features is guided by a predefined criterion of 0.220. After examining the Mendeley DNS Dataset presented in Table 3, a thorough selection procedure resulted in the identification of the 12 most relevant attributes.

4.2.4 Correlation-based feature selection (CFS) technique

Correlation-based Feature Selection is an important approach in machine learning that helps identify the most relevant variables from a dataset. This approach uses correlation to determine the degree of link between characteristics and the target variable, making it easier to pick the features that contribute the most to predictive modelling tasks. CFS is especially useful in situations when datasets include both numerical and categorical information and are used for classification or regression tasks. CFS prioritises features with significant predictive potential and minimises duplication

Feature No.	Information Gain	Feature Name
f1	1.000000000000000256	Domain
f2	0.777969675090184576	NumericSequence
f3	0.75109695664116224	NumericRatio
f4	0.749260387531234944	IP
f5	0.606173981999299456	ASN
f6	0.5782910953568256	StrangeCharacters
f7	0.55113539106572576	ConsoantRatio
f8	0.499455615169670976	DomainLength
f9	0.44302627161621632	SubdomainNumber
f10	0.413287777249203968	VowelRatio
f11	0.229060631434872096	HasSPFInfo
f12	0.22041276495032848	TXTDnsResponse

Table 3: Feature selection using Information Gain method on Mendeley DNS Dataset

among selected characteristics by evaluating their correlation with the target variable [23]. The CFS algorithm generally consists of two basic steps:

- Determine the correlation between each attribute and the target variable.
- Evaluating the redundancy among selected features.

Define X as the feature set, Y as the target variable, and S as the chosen subset of features. Find the correlation between each characteristic X_i and the target variable Y. The Pearson correlation coefficient, which is calculated as:

$$\rho_{X_i,Y} = \frac{\sum_{j=1}^{n}(X_{ij} - \bar{X}_i)(Y_j - \bar{Y})}{\sqrt{\sum_{j=1}^{n}(X_{ij} - \bar{X}_i)^2 \sum_{j=1}^{n}(Y_j - \bar{Y})^2}} \tag{7}$$

where:

- X_{ij} represents the value of feature X_i in the jth observation.
- $\bar{X}_i$ represents the mean of the feature X_i.
- Y_j represents the target variable's value in the jth observation.
- $\bar{Y}$ represents the mean of the target variable.
- n denotes the number of observations.

Rank the features by their correlation coefficients with the target variable.

- Add features to the selected subset S in a stepwise manner.
- At each step, add the feature with the highest correlation coefficient that is not highly correlated with features already in S (to minimize redundancy). One common criterion to measure redundancy is to compute the average pairwise correlation between features in S.
- Repeat step 3 until a stopping requirement is satisfied.

Feature selection prioritises features with better absolute correlation coefficients for predicting the target variable. Positive correlation values imply that when the feature value increases, so does the target variable, whilst negative correlation coefficients indicate the reverse. After calculating correlation coefficients for all characteristics, the top k features with the highest absolute correlation coefficients are selected. The chosen attributes are then used as input for machine learning models. After painstakingly analysing the Mendeley DNS Dataset, as shown in Table 4, selected the four most relevant attributes.

Feature No.	Feature Name
f1	LastUpdateDate
f2	SubdomainNumber
f3	NumericRatio
f4	NumericSequence

Table 4: Feature selection using Correlation-based Feature Selection (CFS) method on Mendeley DNS Dataset

4.3 Machine learning techniques used for malicious domain names detection

Because of their capacity to analyse enormous datasets and uncover patterns indicating harmful behaviour, machine learning techniques have emerged as viable tools for detecting rogue domain names. These methods use various algorithms and models to categorise domain names as benign or dangerous based on attributes gathered from their properties. Supervised learning is a typical machine learning strategy for detecting harmful domain names, in which models are trained on labelled datasets that include both benign and malicious domain names. These models learn to distinguish between the two groups by extracting characteristics including domain age, length, lexical qualities, and historical DNS information. Several supervised learning approaches were used in this work, including AdaBoost, Logistic Regression, K-nearest Neighbours, Naive Bayes, Multilayer Perceptron, and Majority Voting. Additional information on every approach is provided below.

4.3.1 AdaBoost

AdaBoost, or Adaptive Boosting, is a pioneering ensemble learning algorithm commonly used in machine learning for categorization applications. Its innovation stems from its capacity to train a sequence of weak learners consecutively, iteratively concentrating on cases that were misclassified in prior rounds. AdaBoost emphasises difficult-to-classify occurrences by providing larger weights to misclassified examples, resulting in a robust and accurate classification model [24].

AdaBoost is a sort of ensemble learning that combines numerous weak learners to create a robust learner. A mathematical representation of the AdaBoost algorithm is shown here. AdaBoost uses a number of rounds to train weak classifiers and apply weights based on their performance. The final prediction is calculated as the weighted sum of the weak classifiers' predictions. Consider a training dataset written as (X, y), where X signifies the feature vectors and y represents the labels (usually encoded as -1 and 1 for binary classification). The mathematical description of the AdaBoost method is as follows:

Notations:

- D : Dataset with n instances.
- $X = \{x_1, x_2, ..., x_n\}$: Feature vectors.
- $Y = \{y_1, y_2, ..., y_n\}$: Corresponding labels.
- T : Number of boosting rounds (iterations).
- $h_t(x)$: Weak classifier at iteration t.
- α_t : Weight for the weak classifier $h_t(x)$ at iteration t.
- $H(x)$: Final strong classifier.
- w_i^t : Weight of instance x_i at iteration t.
- ϵ_t : Weighted error of weak classifier $h_t(x)$.

Algorithm:

1. Initialize weights: Set initial weights $w_i^1 = \frac{1}{n}, i = 1, 2, ..., n$.
2. For $t = 1$ to T :
   - Train weak classifier: Train weak classifier $h_t(x)$ using weights w_i^t.
   - Calculate error: Calculate weighted error $\epsilon_t = \sum_{i=1}^{n} w_i^t \cdot 1\{h_t(x_i) \neq y_i\}$, where 1 is the indicator function.
   - Calculate weight for classifier: Compute $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
   - Update instance weights: Update weights for next iteration: $w_i^{t+1} = \frac{w_i^t \cdot \exp(-\alpha_t \cdot y_i \cdot h_t(x_i))}{Z_t}$, where Z_t is a normalization factor to ensure weights sum up to 1.
   - Construct final model: $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right)$

AdaBoost offers additional weight to misclassified cases with each iteration, pushing weak classifiers to focus more on them. The weight α_t of each weak classifier is given by its weighted error rate. The final model $H(x)$ combines the weak classifier predictions using weighted majority voting. AdaBoost is a strategy that combines predictions from several weak classifiers, giving preference to those with superior performance. It uses weighted voting to aggregate the individual classifier results, resulting in a final prediction. The main idea behind AdaBoost is to prioritise difficult cases and change data weights during training to highlight misclassified samples. This iterative method is intended to improve the overall performance of the ensemble model.

4.3.2 Logistic Regression

Logistic Regression stands as a fundamental and widely used statistical technique for binary classification tasks. Despite its name, logistic regression is a classification algorithm rather than a regression one. It models the probability of a binary outcome by fitting the data to a logistic function, allowing for efficient estimation of the likelihood of a sample belonging to a particular class. LR is a statistical model used for binary classification. Here's the mathematical representation of logistic regression [25].

Notations:

- m : Number of training examples.
- n : Number of features.
- X : Matrix of input features with dimensions $m \times (n+1)$ (including the intercept term).
- y : Vector of labels with dimensions $m \times 1$.
- θ : Vector of parameters (weights) with dimensions $(n+1) \times 1$.
- $h_\theta(x)$: Hypothesis function for logistic regression.
- $g(z)$: Sigmoid function $g(z) = \frac{1}{1+e^{-z}}$.

.

Algorithm:

1. Initialize Parameters: Initialize the parameter vector θ to zeros or small random values.
2. Hypothesis Function: Define the hypothesis function, $h_\theta(x) = g(\theta^T x)$, where $g(z)$ is the sigmoid function.
3. Cost Function: Define the logistic cost function, $J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$.

4. Gradient Descent:
   - Repeat until convergence: $\theta_j := \theta_j - \alpha\frac{\partial J(\theta)}{\partial\theta_j}$, for $j = 0, 1, ..., n$, where α is the learning rate.
   - Update rule for each parameter θ_j : $\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$
5. Feature Scaling:Scale the features to improve convergence speed if needed.
6. Prediction: Given a new input x, predict the probability that $y = 1$ using $h_\theta(x)$: $h_\theta(x) = g(\theta^T x)$.

Logistic regression uses gradient descent to minimise the logistic cost function and optimise the parameters θ. The sigmoid function $g(z)$ reduces input to the range $(0, 1)$, making it appropriate for describing probabilities. During training, the algorithm iteratively modifies the parameters θ to obtain the appropriate decision boundary. After training, the model may use the hypothesis function $h_\theta(x)$ to estimate the likelihood of an input belonging to the positive class $(y = 1)$.

4.3.3 K-nearest Neighbours

K-nearest Neighbours is a powerful and easy-to-use technique for machine learning classification and regression. It falls under the domain of instance-based learning, in which predictions are produced based on the similarity between the input instance and its neighbours in the feature space. KNN works on the assumption that instances with comparable feature values are likely to belong to the same class or have similar target values. KNN is a basic and effective machine learning technique that may be used for both classification and regression. Here's the mathematical depiction of the KNN algorithm: KNN is a nonparametric, instance-based technique that predicts the majority class or the average of the k-nearest data points in the feature space [26].

Notations:

- D : Training dataset consisting of n instances.
- $X = \{x_1, x_2, ..., x_n\}$: Feature vectors in the dataset.
- K : Number of nearest neighbors to consider.
- x : Input feature vector for which prediction is to be made.

Algorithm:

1. Calculate Distance: For each instance x_i in the training set, Compute the distance between x and x_i using a distance metric (e.g., Euclidean distance, Manhattan distance).
2. Find K Nearest Neighbors: Select the K instances with the smallest distances to x.
3. Majority Vote: For classification, assign the class label that is most frequent among the K nearest neighbors.

KNN is considered a lazy learning method since it does not explicitly construct a model during the training phase. Instead, it saves the whole training dataset. During prediction, it locates the K closest neighbours to the input instance and makes predictions based on them. The distance metric (e.g., Euclidean, Manhattan, Minkowski) and the value of K are critical hyperparameters that determine algorithm performance. Since KNN is sensitive to feature scale, feature scaling is frequently used to normalise the features.

4.3.4 Naive Bayes

Naive Bayes is a basic yet strong probabilistic classification method commonly employed in machine learning due to its efficiency and efficacy in dealing with huge datasets. It is founded on Bayes' theorem, which describes the likelihood of a hypothesis given the data. Despite its "naive" assumption of independence between features, Naive Bayes frequently works extremely well in practice, notably in text classification and spam filtering tasks [27].

Notations:

- D : Dataset with n instances.
- $X = \{x_1, x_2, ..., x_n\}$: Feature vectors in the dataset.
- $Y = \{y_1, y_2, ..., y_n\}$: Corresponding class labels in the dataset.
- x : Input feature vector for which prediction is to be made.
- X_i : Value of the i-th feature.
- $P(X_i|Y)$: Probability of feature X_i given class Y.
- $P(Y)$: Prior probability of class Y.
- $P(Y|X)$: Posterior probability of class Y given features X.

Algorithm:

1. Compute Class Priors: Calculate the prior probabilities $P(Y = y)$ for each class y in the dataset. $P(Y = y) = \frac{\text{number of instances with class } y}{\text{total number of instances}}$.
2. Compute Feature Likelihoods: For each feature X_i and each class y, Calculate the likelihood $P(X_i|Y = y)$ of feature X_i given class y. This can be done using different probability density functions (e.g., Gaussian distribution, multinomial distribution for discrete features).
3. Compute Posterior Probabilities: For a new input feature vector x. For each class y, calculate the posterior probability $P(Y = y|X = x)$ using Bayes' theorem: $P(Y = y|X = x) = \frac{P(Y=y)\cdot\prod_{i=1}^{n}P(X_i=x_i|Y=y)}{\sum_{y'}P(Y=y')\cdot\prod_{i=1}^{n}P(X_i=x_i|Y=y')}$.
4. Predict the Class: Assign the class label $\hat{y}$ with the highest posterior probability: $\hat{y} = \arg\max_y P(Y = y|X = x)$.

Based on the class, Naive Bayes assumes that the features are conditionally independent. This is a strong and frequently impractical assumption, yet it simplifies the computation and typically works well in practice. The approach calculates the class priors (prior probability for each class), the feature likelihoods (conditional probabilities for each feature given each class), and then uses Bayes' theorem to obtain the posterior probabilities. The input feature vector's output class is anticipated to be the one with the highest posterior probability. Naive Bayes is efficient and performs well with high-dimensional data, but it is sensitive to the independence assumption's quality and may overfit with small datasets.

4.3.5 Multilayer Perceptron

The Multilayer Perceptron is a basic artificial neural network design that has been widely applied to machine learning applications such as classification, regression, and pattern recognition. MLPs are made up of several layers of linked neurons, each with one or more neurons that execute calculations on the input data. MLPs are well-known for their capacity to learn complicated patterns from data using the supervised learning method [28]. A Multilayer Perceptron (MLP) is a form of artificial neural network made up of numerous layers of nodes (neurons), with each layer completely linked to the next. Here is the method in several notations:

Notations:

- D : Dataset with n instances.
- $X = \{x_1, x_2, ..., x_n\}$: Feature vectors in the dataset.
- $Y = \{y_1, y_2, ..., y_n\}$: Corresponding labels in the dataset.
- L : Number of layers in the network.
- $\mathbf{W}^{(l)}$: Weight matrix for layer l.
- $\mathbf{b}^{(l)}$: Bias vector for layer l.
- $\mathbf{a}^{(l)}$: Activation vector for layer l.
- $\mathbf{z}^{(l)}$: Pre-activation vector for layer l.
- $\mathbf{y}_{pred}$: Predicted output vector.
- $\sigma(\cdot)$: Activation function, e.g., sigmoid, ReLU, tanh.
- $\mathbf{J}$: Loss function, e.g., cross-entropy for classification, mean squared error for regression.

Algorithm:

1. Initialize Weights and Biases: Initialize the weights $\mathbf{W}^{(l)}$ biases $\mathbf{b}^{(l)}$ for each layer l in the network.
2. Forward Propagation:
   - For each instance x_i in the dataset:
     - Set the input layer's activation $\mathbf{a}^{(1)} = x_i$.
     - For $l = 2, 3, ..., L$:
     - Compute the pre-activation $\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$,
     - Compute the activation $\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)})$.
3. Compute Loss: Compute the loss $\mathbf{J}$ based on the predicted output $\mathbf{y}_{pred}$ and the true labels Y.
4. Backpropagation:
   - Compute the gradient of the loss function with respect to the weights and biases using backpropagation.
   - Update the weights and biases to minimize the loss function using gradient descent or its variants (e.g., Adam, RMSprop): $\mathbf{W}^{(l)} := \mathbf{W}^{(l)} - \alpha \frac{\partial \mathbf{J}}{\partial \mathbf{W}^{(l)}}$, $\mathbf{b}^{(l)} := \mathbf{b}^{(l)} - \alpha \frac{\partial \mathbf{J}}{\partial \mathbf{b}^{(l)}}$ where, α is the learning rate.
5. Repeat steps 2-4 until convergence or for a fixed number of iterations.
6. Prediction: For a new input x , perform forward propagation to compute the predicted output $\mathbf{y}_{pred}$.

Forward propagation is the process of sending input through a network to get the expected outcome. Backpropagation calculates the loss function's gradients with respect to the weights and biases, which are then utilised to update the parameters. This is done iteratively until the model converges or a predetermined stopping threshold is satisfied. MLPs can have numerous hidden layers (thus the term ”multilayer”) with various activation functions in each layer.Training an MLP entails determining the best weights and biases to minimise the loss function, which is commonly accomplished through the use of gradient-based optimisation techniques.

4.3.6 Majority Voting

Majority Voting is a popular ensemble learning approach that combines predictions from numerous base classifiers to increase overall prediction accuracy and resilience. It works on the premise that combining the decisions of numerous classifiers can result in higher performance than any single classifier alone. Majority Voting is especially useful when the basis classifiers are varied and produce uncorrelated mistakes. Majority voting is a basic ensemble approach in which the final prediction is chosen by the majority of individual classifier votes. Here's the algorithm with several notations [29, 30, 31, 32, 33].

Notations:

- C : Set of classifiers.
- n : Number of classifiers in C.
- y_i : Predicted label of the i-th classifier.

Algorithm:

1. Predictions: For each instance in the dataset, each classifier in C predicts the label for the instance.
2. Majority Vote:
   - For each instance:
   - Count the number of votes for each class label across all classifiers.
   - Choose the class label with the highest count as the final prediction.
   - In case of ties, break the tie using a predefined rule (e.g., randomly, based on class probabilities, etc.).

Classification problems are generally solved by majority vote. The label for each event is predicted individually by each classifier in the ensemble. The final prediction for each instance is made by picking the class label with the highest votes from all classifiers. Majority voting may be applied to any classifier, such as decision trees, logistic regression, support vector machines, and so on. It is a basic but effective ensemble approach that frequently increases classification accuracy, particularly when the various classifiers have varying predictions. Majority voting may alternatively be converted into soft voting, in which the probabilities predicted by each classifier are averaged rather than calculating the hard votes.

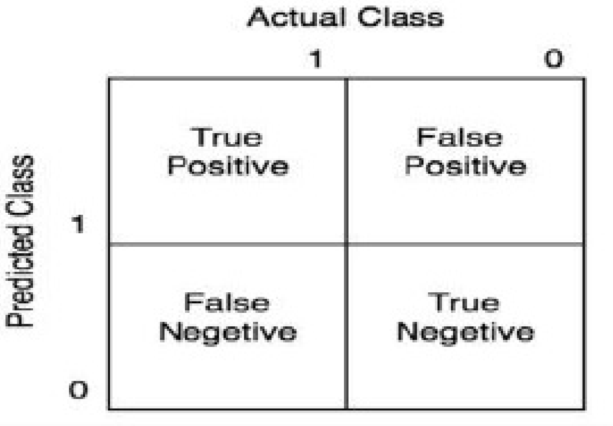
5 Experimental results

5.1 Dataset

This study uses the Mendeley Dataset, a large database of benign and malicious domains retrieved from DNS records [18]. This dataset, specifically designed for supervised machine learning research targeted at distinguishing between dangerous and non-malicious domain names, was methodically produced by pooling publically available DNS logs belonging to both types of domain names. Each domain name in the dataset serves as input, yielding a total of 34 characteristics. These characteristics include direct attributes taken from domain names, such as entropy, the occurrence of uncommon characters, and domain length. Additional information, such as domain creation date, related IP address, open ports, and geolocation, were gathered using data enrichment methods that employed Open Source Intelligence methodologies. This dataset, consisting of 90,000 domain names, is carefully balanced to provide an equal distribution of 50% non-malicious and 50% malicious domains. During the preprocessing step, five characteristics are omitted from both the training and testing datasets: DNSRecordType, CountryCode, RegisteredCountry, RegisteredOrg, and TLD. As a consequence, the final dataset used for experimentation has 29 characteristics that serve as inputs for model training and testing.

5.2 Measures used for performance evaluation of learning classifiers on mendeley DNS dataset

The usefulness of different machine learning models, including AdaBoost, Logistic Regression, K-Nearest Neighbours, Naive Bayes, Multilayer Perceptron, and Majority Voting, in detecting fraudulent domain names was investigated. The following metrics are used to measure the proposed approach's success in detecting malicious domain names. Figure 3 depicts the confusion matrix, which includes True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These criteria, which provide a measurable evaluation of accuracy, precision, recall, and overall performance, are often used to evaluate the success of classification algorithms [34].



Predicted Class \ Actual Class	1	0
1	True Positive	False Positive
0	False Negetive	True Negetive

Figure 3: Confusion Matrix for malicious domain names detection

- **Accuracy:** Accuracy is a statistic that measures a classification model's overall efficacy. It expresses the proportion of accurately predicted cases to the total number of examples in the dataset. It is computed using the equation shown below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

- **Precision:** Precision evaluates the accuracy of the model's favourable predictions. It is computed as the ratio of genuine positive predictions to total positive forecasts.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

- **Recall:** Recall measures the model's ability to properly identify positive cases. It is the ratio of genuine positive forecasts to all instances of positive events. It is computed using the equation shown below.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

- **F-Measure (F1-Score):** TThe F1-score, which is the harmonic mean of accuracy and recall, gives a fair evaluation of the model's performance. It is computed using the equation shown below.

$$F1\text{-}Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{11}$$

5.3 Performance evaluation of machine learning classifiers on Mendeley DNS dataset using full feature set

Table 5 compares the performance of each machine learning classifier against a comprehensive feature set consisting of 29 characteristics from the Mendeley DNS Dataset. AdaBoost outperformed other classifiers, with 99.96% detection accuracy, precision, recall, and F-measure. Furthermore, the AdaBoost classifier has fast training and testing durations, taking 3.581 seconds to train and 0.225 seconds to test. Figure 4 and 5 shows the performance evaluation and ROC curve for the classifiers.

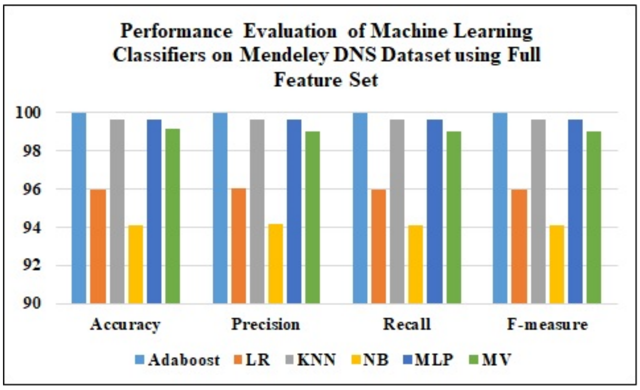


Figure 4: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Full Feature Set

5.4 Performance evaluation of machine learning classifiers on Mendeley DNS dataset using Chi-square (χ^2) feature selection technique

Table 6 demonstrates the performance evaluation of individual machine learning classifiers on the Mendeley DNS Dataset, using 12 features and the Chi-square Feature Selection Method. The AdaBoost, Naive Bayes, Multilayer Perceptron, and Majority Voting classifiers outperformed each other in detecting malicious DNS, as indicated by better accuracy, precision, recall, and F-measure. This increase is accomplished while employing only 12 features as opposed to the complete set of 29 features, resulting in low system overhead. The AdaBoost classifier obtained 99.97% accuracy, precision, recall, and F-measures. Similarly, the KNN classifier achieved 99.67% accuracy, precision, recall, and F-measure. Figure 6 and 7 illustrates the classifiers' performance evaluation and ROC curves.

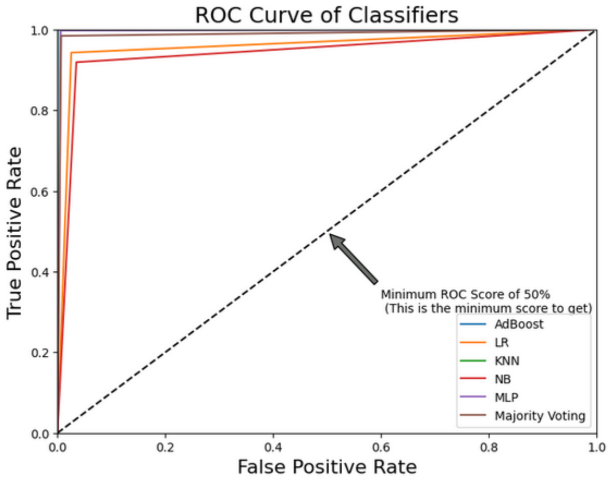


Figure 5: ROC Curve of Machine Learning Classifiers on Mendeley DNS Dataset using Full Feature Set

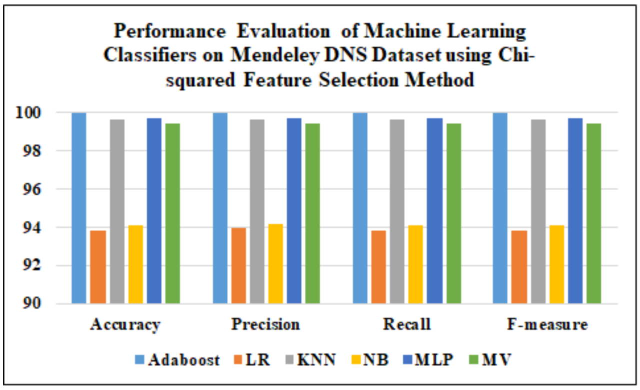


Figure 6: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Chi-squared (χ^2) Feature Selection Technique

5.5 Performance evaluation of machine learning classifiers on Mendeley DNS dataset using GainRatio feature selection technique

Table 7 compares the performance of individual machine learning classifiers on the Mendeley DNS Dataset using 12 features and the GainRatio Feature Selection technique. The AdaBoost classifier had the maximum detection accuracy, precision, recall, and F-measure (all at 96.96%). Most classifiers showed gains in training and testing time, with the exception of the Multilayer Perceptron and Majority Voting classifier. Figure 8 and 9 shows the performance evaluation and the ROC curve for the classifiers.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Train Time (s)	Test Time (s)
AdaBoost	99.96	99.96	99.96	99.96	3.581	0.225
LR	96	96.04	96	96	1.035	0.0103
KNN	99.67	99.67	99.67	99.67	0.0195	6.337
NB	94.06	94.15	94.06	94.06	0.04	0.012
MLP	99.61	99.61	99.61	99.61	92.504	0.032
MV	99.15	99	99	99	106.222	5.003

Table 5: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Full Feature Set

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Train Time (s)	Test Time (s)
AdaBoost	99.97	99.97	99.97	99.97	2.73	0.186
LR	93.81	93.97	93.81	93.81	0.757	0.007
KNN	99.67	99.67	99.67	99.67	0.144	2.957
NB	94.1	94.15	94.1	94.1	0.0246	0.007
MLP	99.74	99.74	99.74	99.74	67.467	0.0456
MV	99.44	99.44	99.44	99.44	48.468	5.299

Table 6: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Chi-square (χ^2) Feature Selection Technique

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Train Time (s)	Test Time (s)
AdaBoost	96.96	96.96	96.96	96.96	2.247	0.1889
LR	90.01	90.37	90.01	89.99	0.8859	0.0071
KNN	94.44	94.52	94.44	94.44	0.1465	6.3003
NB	90.62	90.74	90.62	90.61	0.0244	0.00688
MLP	97.9	97.9	97.9	97.9	123.188	0.058
MV	96.55	96.55	96.55	96.55	112.42	6.377

Table 7: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using GainRatio Feature Selection Technique

5.6 Performance evaluation of machine learning classifiers on Mendeley DNS dataset using information gain feature selection technique

Table 8 shows the performance evaluations of individual machine learning classifiers on the Mendeley DNS Dataset, using 12 features and the Information Gain Feature Selection Method. The AdaBoost, KNN, Multilayer Perceptron, and Majority Voting classifiers all performed better at detecting malicious DNS. This improvement is demonstrated by higher accuracy, precision, recall, and F-measure measures. Interestingly, these classifiers achieve these results while using only 12 characteristics rather than the entire set of 29, reducing system overhead. The AdaBoost classifier performs with 99.97% accuracy, precision, recall, and F-measure. Figure 10 and 11 shows the performance evaluation and the ROC curve for the classifiers.

5.7 Performance evaluation of machine learning classifiers on Mendeley DNS dataset using correlation based feature selection (CFS) technique

Table 9 compares the performance of individual machine learning classifiers on the Mendeley DNS Dataset, using 4 features and the Correlation-based Feature Selection approach. The AdaBoost classifier outperformed others with 94.91% detection accuracy, 95.10% precision, 94.91% recall, and 94.91% F-measure. Except for the Multilayer Perceptron classifier, most classifiers improved their training and testing times. Figure 12 and 13 shows the performance evaluation and ROC curve for the classifiers.

5.8 Comparative performance evaluation with available approaches

To verify the effectiveness of the proposed approach comparative performance evaluation is performed with other available approaches [3, 5, 6, 9, 16]. Table 10 shows the comparative performance evaluation of our proposed approach with available approaches in terms of accuracy, precision, recall and F-measure.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Train Time (s)	Test Time (s)
AdaBoost	99.97	99.97	99.97	99.97	2.7555	0.19073
LR	92.96	93.07	92.96	92.95	0.472	0.007
KNN	99.69	99.69	99.69	99.69	0.144	2.897
NB	94.04	94.12	94.04	94.04	0.0235	0.006
MLP	99.75	99.75	99.75	99.75	97.564	0.0555
MV	99.4	99.4	99.4	99.4	61.0559	5.264

Table 8: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Information Gain Feature Selection Technique

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)	Train Time (s)	Test Time (s)
AdaBoost	94.91	95.10	94.91	94.91	1.221	0.1704
LR	93.13	93.13	93.13	93.13	0.8254	0.0045
KNN	94.16	94.16	94.16	94.16	0.0513	5.913
NB	91.68	91.68	91.68	91.68	0.0182	0.0041
MLP	95.41	95.52	95.41	95.41	101.938	0.1046
MV	94.73	94.73	94.73	94.73	62.4239	4.751

Table 9: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Correlation based Feature Selection (CFS) Technique

Approach	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
Zhao, Hong et al. [3]	94.04	–	–	–
Luhui Yang et al. [5]	89.20	93.20	84.07	88.40
Shaojie Chen et al. [6]	97.24	97.22	97.25	97.23
Zhaoshan Fan et al. [9]	99.61	87.76	96.84	92.08
Vinayakumar, R. et al. [16]	97.4	87.8	99.3	93.2
Proposed Approach	99.44	99.44	99.44	99.44

Table 10: Comparative Performance Evaluation with Available Approaches

5.9 Limitations

This paper proposes a method for detecting malicious domain names using a combination of feature selection and a majority voting approach among different classifiers. Here are some potential limitations of such a framework:

- **Feature Dependence:** The effectiveness of the model heavily relies on the quality and relevance of the selected features. If important features are omitted or irrelevant features are included, the detection performance may degrade.
- **Class Imbalance:** Malicious domain datasets often suffer from class imbalance, where the number of benign domains significantly outweighs the number of malicious ones. This can lead to biased models that perform well on benign domains but poorly on malicious ones.
- **Evasion Techniques:** Attackers continually develop new evasion techniques to avoid detection. A framework based on static features may become less effective over time as attackers adapt their strategies.
- **Data Quality and Availability:** The availability of high-quality labeled data for training and testing the models is crucial. In practice, obtaining such datasets can be difficult, and models trained on outdated or incomplete data may not generalize well to new threats.
- **Dynamic Nature of Domains:** Domain names can change rapidly, and new malicious domains can appear at any time. The framework needs to be regularly updated to maintain its effectiveness.

6 Conclusions

This paper describes a method for identifying harmful domain names that combines feature selection approaches with a majority vote approach. Several approaches, including Chi-square, Gain Ratio, Information Gain, and Correlation-based feature selection, are used to discover the most relevant information for detecting harmful domain names while minimising repetition. The experimental results demonstrate the usefulness of the proposed framework, notably the majority voting strategy, which uses several classifiers to improve detection accuracy. The results show impressive performance metrics, including accuracy, precision, recall, and F-measure, which are more acceptable compared to individual classifiers, of 99.44%. This

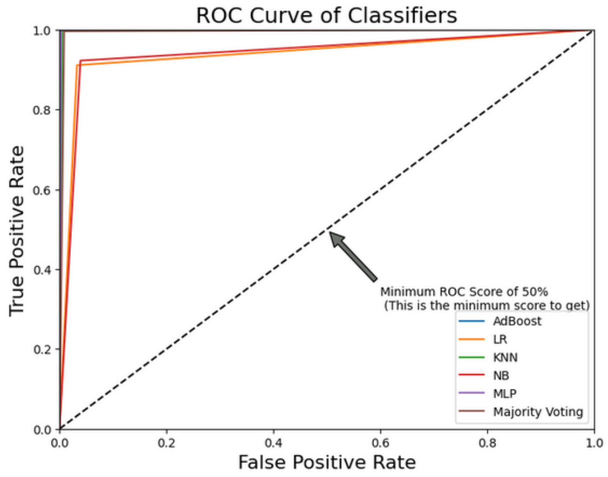


Figure 7: ROC Curve for classifiers using Chi-squared (χ^2) Feature Selection Technique

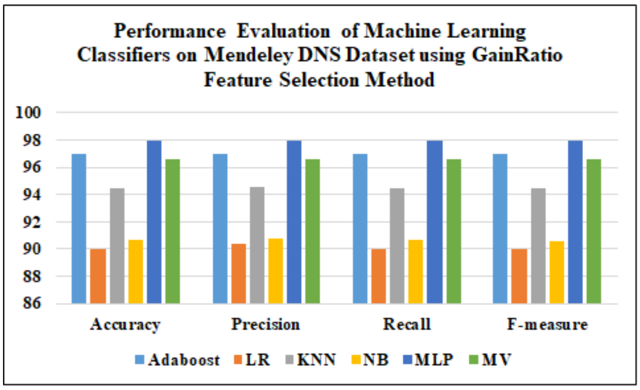


Figure 8: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using GainRatio Feature Selection Technique

was achieved through the combination of Majority Voting and Chi-squared Feature Selection Method, using only 12 domain name features. This demonstrates the framework's ability to correctly identify malicious domain names while minimising system overhead. Furthermore, a comparison of various feature selection approaches and machine learning classifiers provides useful insights into the best combinations for obtaining higher detection performance. The framework's adaptability to various forms of malicious domain names is proved by thorough experimentation on real-world datasets.

List of abbreviations

The following abbreviations are used in this manuscript:

URL Uniform Resource Locator

AI Artificial intelligence

DNS Domain Name System

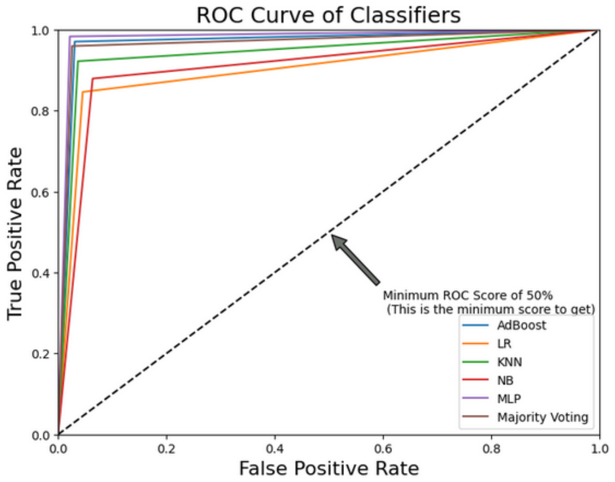


Figure 9: ROC Curve of classifiers using GainRatio Feature Selection Technique

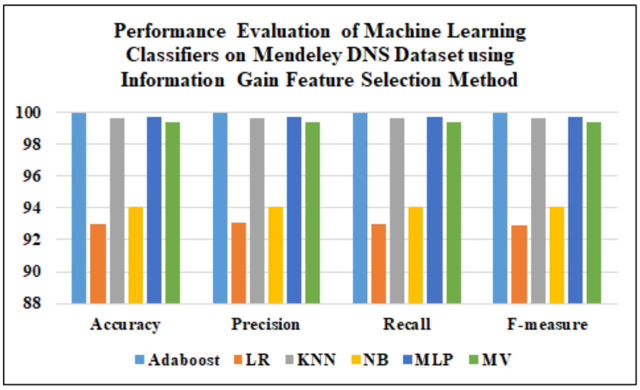


Figure 10: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Information Gain Feature Selection Technique

PCA Principal Component Analysis

AGD Algorithmically Generated Domains

PUMD Positive and Unlabeled Malicious Domain

DGA Domain Generation Algorithms

ELM Extreme Learning Machine

KL Kullback-Leibler

ED Edit Distance

JI Jaccard Index

HMM Hidden Markov Models

PCFG Probabilistic Context-Free Grammars

LSTM Long short-term memory

AUC Area under the ROC Curve

HDNN Heterogeneous Deep Neural Network

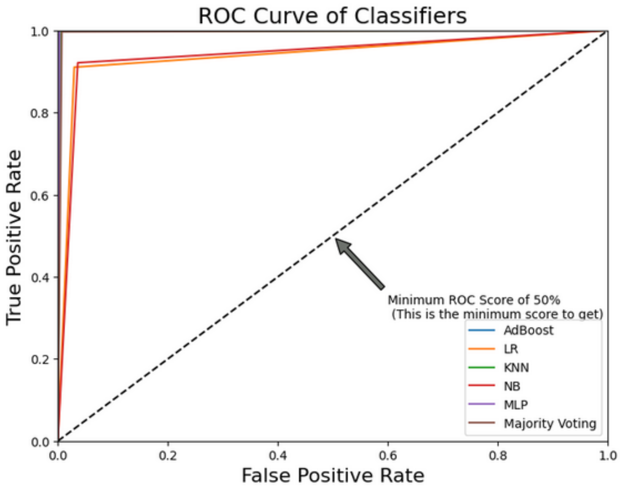


Figure 11: ROC Curve for classifiers using Information Gain Feature Selection Technique

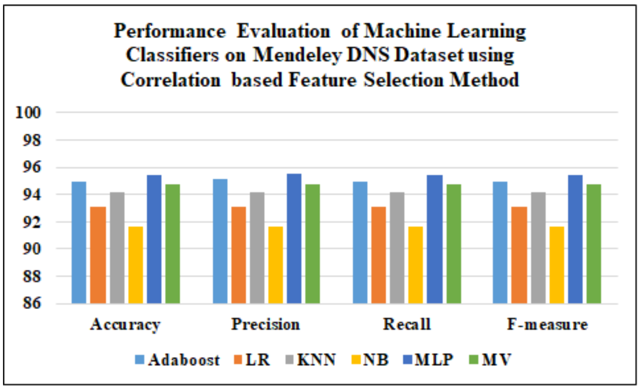


Figure 12: Performance Evaluation of Machine Learning Classifiers on Mendeley DNS Dataset using Correlation based Feature Selection (CFS) Technique

SDGA Sub-domain Domain Generation Algorithm

LAN Local Area Network

RNN Recurrent Neural Networks

IP Internet Protocol

TLD Top-level Domain

IGR Information Gain Ratio

IG Information Gain

CFS Correlation-based Feature Selection

AdaBoost Adaptive Boosting

LR Logistic Regression

KNN K-nearest Neighbours

NB Naive Bayes

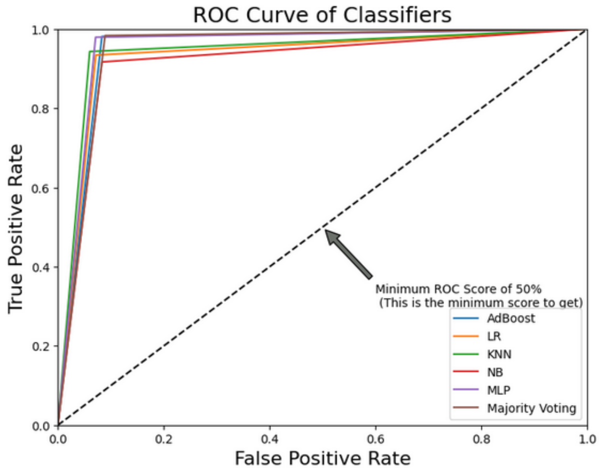


Figure 13: ROC Curve for classifiers using Correlation based Feature Selection (CFS) Technique

MLP Multilayer Perceptron

MV Majority Voting

TP True Positives

TN True Negatives

FP False Positives

FN False Negatives

ROC Receiver Operating Characteristic

References

[1] Interisle malicious domain names statistics 4Q 2022. Available online, `https://www.cybercrimeinfocenter.org/malware-landscape-2023`.

[2] CSC domain security 2023 report. Available online, `https://www.cscdbs.com/assets/pdfs/2023-Domain-Security-Report.pdf`.

[3] Zhao, Hong, Zhaobin Chang, Guangbin Bao, and Xiangyan Zeng, Malicious domain names detection algorithm based on N-gram. *Journal of Computer Networks and Communications 2019*. `https://doi.org/10.1155/2019/4612474`.

[4] Soleymani, Ali, and Fatemeh Arabgol, A novel approach for detecting DGA-based botnets in DNS queries using machine learning techniques. *Journal of Computer Networks and Communications*, 2021, 1–13.`https://doi.org/10.1155/2021/4767388`.

[5] Yang, Luhui, Guangjie Liu, Weiwei Liu, Huiwen Bai, Jiangtao Zhai, and Yuewei Dai,Detecting Multielement Algorithmically Generated Domain Names Based on Adaptive Embedding Model, *Security and Communication Networks*, 2021, 1–20. https://doi.org/10.1155/2021/5567635.

[6] Chen, Shaojie, Bo Lang, Yikai Chen, and Chong Xie, Detection of Algorithmically Generated Malicious Domain Names with Feature Fusion of Meaningful Word Segmentation and N-Gram Sequences, *Applied Sciences*, 13, no. 7,2023, 4406. https://doi.org/10.3390/app13074406.

[7] Wagan, Atif Ali, Qianmu Li, Zubair Zaland, Shah Marjan, Dadan Khan Bozdar, Aamir Hussain, Aamir Mehmood Mirza, and Mehmood Baryalai, A Unified Learning Approach for Malicious Domain Name Detection, *Axioms*, 12, no. 5, 2023, 458. https://doi.org/10.3390/axioms12050458.

[8] Bilge, Leyla, Engin Kirda, Christopher Kruegel, and Marco Balduzzi, Exposure: Finding malicious domains using passive DNS analysis, *In Ndss*, pp. 1–17, 2011.

[9] Fan, Zhaoshan, Qing Wang, Haoran Jiao, Junrong Liu, Zelin Cui, Song Liu, and Yuling Liu, PUMD: a PU learning-based malicious domain detection framework, *Cybersecurity*, 5, no. 1, 2022, 1–22. https://doi.org/10.1186/s42400-022-00124-x.

[10] Yang, Luhui, Jiangtao Zhai, Weiwei Liu, Xiaopeng Ji, Huiwen Bai, Guangjie Liu, and Yuewei Dai, Detecting word-based algorithmically generated domains using semantic analysis, *Symmetry*, 11, no. 2, 2019, 176. https://doi.org/10.3390/sym11020176.

[11] Shi, Yong, Gong Chen, and Juntao Li, Malicious domain name detection based on extreme machine learning, *Neural Processing Letters*, 48,2018,1347–1357. https://doi.org/10.1007/s11063-017-9666-7.

[12] Fu, Yu, Lu Yu, Oluwakemi Hambolu, Ilker Ozcelik, Benafsh Husain, Jingxuan Sun, Karan Sapra, Dan Du, Christopher Tate Beasley, and Richard R. Brooks, Stealthy domain generation algorithms, *IEEE Transactions on Information Forensics and Security*, 12, no. 6, 2017, 1430–1443. https://doi.org/10.1109/TIFS.2017.2668361.

[13] Yun, Xiaochun, Ji Huang, Yipeng Wang, Tianning Zang, Yuan Zhou, and Yongzheng Zhang, Khaos: An adversarial neural network DGA with high anti-detection ability, *IEEE transactions on information forensics and security*, 15, 2019,, 2225–2240. https://doi.org/10.1109/TIFS.2019.2960647.

[14] Yang, Luhui, Guangjie Liu, Yuewei Dai, Jinwei Wang, and Jiangtao Zhai, Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework, *IEEE Access*, 8, 2020,82876–82889. DOI: 10.1109/ACCESS.2020.2988877.

[15] Xu, Congyuan, Jizhong Shen, and Xin Du, Detection method of domain names generated by DGAs based on semantic representation and deep neural network, *Computers & Security*, 85, 2019,77–88. https://doi.org/10.1016/j.cose.2019.04.015.

[16] Vinayakumar, R., K. P. Soman, and Prabaharan Poornachandran, Detecting malicious domain names using deep learning approaches at scale, *Journal of Intelligent & Fuzzy Systems*, 34, no. 3, 2018,1355–1367. https://doi.org/10.3233/JIFS-169431.

[17] Yang, Luhui, Guangjie Liu, Jinwei Wang, Jiangtao Zhai, and Yuewei Dai, A semantic element representation model for malicious domain name detection, *Journal of Information Security and Applications*, 66, 2022,103148. https://doi.org/10.1016/j.jisa.2022.103148.

[18] Marques, Claudio, Benign and malicious domains based on DNS logs, *Mendeley Data*, V5, 2021, doi: 10.17632/623sshkdrz.5.

[19] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH, The WEKA data mining software: an update, *ACM SIGKDD explorations newsletter*, 2009, Nov 16, 11(1),10–8.

[20] Zhai Y, Song W, Liu X, Liu L, Zhao X, A chi-square statistics based feature selection method in text classification, *In 2018 IEEE 9th International conference on software engineering and service science (ICSESS)*, 2018, Nov 23,pp. 160–163, IEEE.

[21] Prasetiyo B, Muslim MA, Baroroh N, Evaluation of feature selection using information gain and gain ratio on bank marketing classification using Naïve bayes, *In Journal of physics: conference series*, 2021, Jun 1,Vol. 1918, No. 4, pp. 042153, IOP Publishing. DOI: 10.1088/1742-6596/1918/4/042153

[22] Qu K, Xu J, Hou Q, Qu K, Sun Y., Feature selection using Information Gain and decision information in neighborhood decision system, *Applied Soft Computing*, 2023, Mar 1, 136,110100. https://doi.org/10.1016/j.asoc.2023.110100.

[23] Hall, Mark A., Correlation-based feature selection of discrete and numeric class machine learning, 2000.

[24] Schapire RE., Explaining AdaBoost, *In Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, 201,3 Oct 9, pp. 37–52,. Berlin, Heidelberg: Springer Berlin Heidelberg.

[25] Stoltzfus JC., Logistic regression: a brief primer, *Academic emergency medicine*, 2011, Oct, 18(10), 1099–104. https://doi.org/10.1111/j.1553-2712.2011.01185.x.

[26] Peterson LE., K-nearest neighbor, *Scholarpedia*, 2009, Feb 21, 4(2),1883.

[27] Rish, Irina., An empirical study of the naive Bayes classifier, *In IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22, pp. 41–46. 2001.

[28] Tang, Jiexiong, Chenwei Deng, and Guang-Bin Huang, Extreme learning machine for multilayer perceptron, *IEEE transactions on neural networks and learning systems*, 27, no. 4, 2015, 809–821. https://doi.org/10.1109/TNNLS.2015.2424995.

[29] Ruta D, Gabrys B., Classifier selection for majority voting, *Information fusion*, 2005, Mar 1, 6(1), 63-81. https://doi.org/10.1016/j.inffus.2004.04.008.

[30] Patil, Dharmaraj R., Tareek M. Pattewar, Vipul D. Punjabi, and Shailendra M. Pardeshi, Detecting Fake Social Media Profiles Using the Majority Voting Approach, *EAI Endorsed Transactions on Scalable Information Systems*, 2024. https://doi.org/10.4108/eetsis.4264.

[31] Patil, Dharmaraj R., and Tareek M. Pattewar, Majority Voting and Feature Selection Based Network Intrusion Detection System, *EAI Endorsed Transactions on Scalable Information Systems*, 9, no. 6,2022: e6-e6. https://doi.org/10.4108/eai.4-4-2022.173780.

[32] Patil, Dharmaraj R., Fake news detection using majority voting technique, *arXiv preprint*, arXiv:2203.09936, 2022. https://doi.org/10.48550/arXiv.2203.09936.

[33] Patil, Dharmaraj R., and Jayantro B. Patil, Malicious URLs detection using decision tree classifiers and majority voting technique, *Cybernetics and Information Technologies*, 18, no. 1, 2018: 11-29. DOI: 10.2478/cait-2018-0002.

[34] Sokolova M, Lapalme G., A systematic analysis of performance measures for classification tasks, *Information processing & management*, 2009, Jul 1, 45(4), 427–37. https://doi.org/10.1016/j.ipm.2009.03.002.

CerConvNet: Cervical Cancer Cells Prediction Using Convolutional Neural Networks

Pallavi M, Smitha Patil, Madhusudhan M V, Smitha S Reddy, Vaishnavi K
Presidency University, Bangalore
E-mail: pallavim@presidencyuniversity.in, smithapatil@presidencyuniversity.in,
madhusudhanmv@presidencyuniversity.in, smithasreddy13@gmail.com, vaishnavikosgi6772@gmail.com

Keywords: deep learning, CNN, SIPaKMeD, cervix.

Received: March 14, 2024

Cervix cancer is a distinct form of cancer occurring in women, originating in the cells of the cervix, which is the region of the uterus connecting to the vagina. About 90% of cases of cervix cancer are related to human papillomavirus (HPV) infection. The mortality rate in developed nations has decreased because of routine HPV testing for women. The absence of reasonably priced healthcare facilities, however, continues to make it difficult for developing countries to offer inexpensive remedies. Therefore, developing an accurate algorithm for cervical cancer prediction is necessary to identify women who are at risk of developing this condition. Architectures of Deep Learning have been employed in recent years to construct accurate models for the prediction of cervical cancer. This study offers a unique, straightforward transfer learning framework: ResNet50, DenseNet201, EfficientNetb1 and InceptionResNetV2, to classify cervical images using the SIPaKMeD dataset and different performance measures are gathered and examined. Still, the recommended Densenet201outperformed the most advanced methods. We obtained an average accuracy of 98.78% with CNN models which is the highest compared over the existing models. Resnet50 achieved even better results after augmentation with an accuracy of 99.51% and Precision, Recall, F1-score of 0.99. As a result, the findings support our approach to providing low-cost first-level screening.

Povzetek: Raziskava predstavlja CerConvNet, okvir za napovedovanje raka materničnega vratu z uporabo konvolucijskih nevronskih mrež. Model DenseNet201 je dosegel najvišjo točnost, izboljšano z bogatenjem podatkov.

1 Introduction

In recent years, deep learning (DL), a sector of Artificial Intelligence (AI), has experienced rapid and substantial growth. The field of science is currently focusing on deep learning (DL) because of its many benefits, including high performance, diverse applications, great generalisation capability, and versatility. In view of the processing of huge amount of medical data and the advancement of computing capacity have created an interest in this domain. Utilizing deep learning could expedite and enhance the diagnostic process, leading to more precise and personalized treatment approaches. The study of computer-aided medical picture analysis holds great potential for bettering patient results [1]

As of 2020, cervical cancer ranks as the fourth most prevalent cancer among women, one in 168 Canadian women is predicted to get cervical cancer in their lives, and one in 478 Canadian women passes away from the disease. Cervical cancer is most widespread and fatal in countries with lower and middle incomes. The disparities emerge because of social and economic elements, along with restricted availability of nationwide HPV vaccination, cervical screening, and treatment resources, highlighting notable inequalities. The cause for cervical cancer is infection with high-risk variants of human papillomavirus (HPV). These disparities stem from social and economic factors, along with restricted availability of national HPV vaccination, cervical screening, and treatment resources, highlighting substantial inequities. The timely identification and prompt treatment are crucial elements in successfully curing cervical cancer. The cofactors that may increase the risk are tobacco smoking, immune suppression, multi-parity, sexually transmitted infections and poor diet. Lack of regular screening also increases the risk of Cervical cancer. The common symptoms include irregular or heavy bleeding, bleeding after intercourse, unpleasant odor in vaginal discharge and discomfort during sexual activity. Cervix cancer can be diagnosed by history or recto vaginal examination. The size of the tumour and the spread of the disease determine the cancer stage. In the approaching decades, countries globally are working towards accelerating the elimination of cervical cancer, aiming to achieve a specific set of three objectives by 2030.

2 Related work

Using DL algorithms, several previous research have suggested methods for detecting and classifying CC. Wanli Liu, Chen Li, Ning Xu et al. suggested a deep learning based image classification model known as CVM-Cervix to perform cervix cell classification [1]. CVM-Cervix is evaluated using a merged dataset that combines CRIC and SIPaKMeD datasets, constituting 11 classes. The pre-processed data is fed into the Convolutional Neural Networks and Virtualisation Technology modules for feature extraction. The features extracted are then supplied into the MLP module for fusion and classification. Additionally, a streamlined post-processing technique is used to improve the model. The performance of CVM-Cervix is compared with 22 models (18 CNN models, 4 VT models). Among the 18 CNN models, high accuracy is shown by DenseNet169(88.99%) and DeiT (87.35%) has the highest accuracy among the VT models. However, CVM-Cervix surpassed both with an accuracy of 91.72%.

Anurag Tripathi et al. presents a deep learning approach on the SIPAKMED dataset using the ResNET-152 Architecture. This project involves classifying images into five groups. Normal cells and abnormal cells are the two subclasses that are created from the five classes. Transfer learning is implemented. The dataset was trained with 4 models, namely ResNet50, ResNet152, VGG16 and VGG19. All models were adjusted using identical hyper-parameter conFigureurations. The fine-tuning of model weights was performed over 50 scans over the entire dataset, considering a batch-size of 10, input image resolution set to 224×224. ResNet-152 yielded the highest accuracy of 94.89%, followed by VGG19(94.38%), ResNet-50(93.87%) and VGG16(92.85).

Swati Shinde, Madhura Kalbhor et al. presented an approach for prediction [2]. It introduces a deep learning framework DeepCyto, developed using pre-trained CNN models and ANN model. This frame work was tested on 3 cervical cancer datasets (Herlev, Sipakmed and LBC). In workflow 1, the feature vectors extracted from Principal Component analysis (PCA) are fed into 3 Machine learning classifiers (SVM, RF and FCN), the classifier's prediction is then passes into the voter. The maximum voting process's result determines the image's anticipated class. In workflow 2, the deep features are extracted using 4 CNN pre-trained models (ResNet50, XceptionNet, VGG16, VGG19), these features are then passed into a fully connected ANN. The pre-trained models' feature fusion vector is used to train the model, and the Artificial Neural Network's (ANN) output layer estimates the probability outputs that are used to make subsequent predictions. When the two workflows were compared, workflow two showed better performance. Achieving accuracy of 97-100% on different datasets.

Wasswa William, Andrew Ware et al. proposes a framework of automated cervical cancer diagnosis and categorisation. [3]. The suggested tool includes scene segmentation through the utilization of categoriser named WEKA region of interest and utilizes a method of systematically eliminating one item at a time to discard unwanted debris. Fuzzy C-means technique is used for classification, and simulated annealing is integrated with a wrapper filter for feature selection. The findings indicated that the approach surpasses numerous existing algorithms in terms of sensitivity is 99.28%, specificity of 97.47%, and accuracy is about 98.88% when tested on the Herlev dataset. The suggested system examines an entire pap smear slide in 3 minutes, contrasting with the 5–10 minutes required for manual analysis per slide.

Alquran, H, Alsalatie et al. proposes Cervical Net, a novel DL structure with distinguished group convolutional layers, utilizing depth-wise separable convolutions and grouped convolutions to extract depth features for improved accuracy in cervical cancer classification [4]. The model comprises stages such as image acquisition, image enhancement, extracting and choosing features, combining features, and carrying out classification. ShuffleNet V2 addresses the challenge of computational complexity in the model by using depth-wise convolutions and 1x1 tiny convolution kernels, resulting in a smaller model size without compromising accuracy. The extracted features from Cervix Net are input to different ML algorithms (SVM, RF, ANN Naïve Bayes, KNN) to determine the optimal classifier accuracy. These classifiers were verified on the SIPaKMeD datasets. The system attained its greater accuracy for 5 classes with 99.1 percent using the Support Vector Machine, while the Naïve-Bayes'classifier yielded lesser results, not surpassing eighty-five percent of accuracy.

Taranjit Kaur, Tapan Kumar Gandh et al. suggests the classification of brain images employing Convolutional Neural-Networks (CNN) and the application of transfer learning techniques [5]. Utilized CNN architectures such as Alexnet, ResNet-50, GoogLenet, VGG16, ResNet-101, VGG19, Inception-V3, and InceptionResNet-V2. It makes use of transfer learning - to increase the efficiency of the model, it makes use of the initial layers of pre-trained model by replacing last layers of the network. The model is evaluated on the images taken from the Harvard repository, Figure share repository, and Fortis Memorial Research Institute. The evaluation metrics used are sensitivity, specificity, and accuracy. The results showed AlexNet model yielded the most favorable results for all three datasets, with VGG-16 closely trailing behind. Employing pre-trained Deep-Convolutional-Neural-Network (DCNN) architectures with knowledge transfer technique for classification presents several advantages. Firstly, it automates the entire classification process. Secondly, it removes the traditional processes of noise filtering, outlining Regions of Interest (ROI), and conducting feature extraction and selection. Thirdly, it eliminates biases both between different observers and within the same observer, ensuring reproducible predictions from the pre-trained DCNN models.

Pin Wang, Lirui Wang et al. The suggested approach employs the Mean-shift clustering algorithm for identifying Regions of Interest (ROI). Subsequently, a versatile mathematical morphological operation is employed to separate nuclei that may be overlapping [6]. The project involves two steps, the first step is to accurately segment the cell nuclei present in the Pap smear images. This is achieved through Mean-Shift Clustering Algorithm and Mathematical Morphology. After segmenting the cell nuclei, the next step is to categorise samples of images based on the characteristics of the nuclei. Features derived from shape, texture, and Gabor are extracted from the segmented nuclei for classification. The proposed segmentation and classification methods are tested on a database of 362 cervical Pap smear images. The proposed system demonstrates high effectiveness in segmenting cell nuclei, with a sensitivity of approximately

94.25% and specificity of about 93.45% and an accuracy of more than 96%.

Md Mamunur Rahaman, Chen Li, Xiangchen Wu et al. proposed a model which includes, gathering image data, Image data cleansing, ROI of image extraction and categorisation of pictures using deep learning techniques, creating feature vectors and classifiers [7]. Three kinds of feature learnings, namely unsupervised (Autoencoders, Sparse and topological, etc), Supervised (ResNet, Fully CNN, VGG, Patch based, etc), Hybrid (Unsupervised pre-trained + CNN, CNN+ graph-based method, etc) are used for segmentation tasks. Classification using deep learning involves both representation learning techniques (such as Deep CNN, VGG16, AlexNet, ResNet, etc.) and cross representation learning techniques (including CNN + transfer learning, AlexNet + SVM, CNN + feature concatenation, etc.) [8]. CNN is acknowledged for its exceptional performance in both segmentation and classification tasks. It later explains a wide variety of CNN architectures namely AlexNet, ResNet, VGGNet, Inception, and mentions the available public and private cervical cytology datasets.

Pin Wang, En Fan, Peng Wang et al. contrasts and examines older machine learning and deep learning techniques for image categorisation, employing the SVM and CNN algorithms for comparison. The experiment is done on the MNIST hand-written digital picture dataset and COREL1000 picture set [9]. The article illustrates a substantial MNIST dataset, Support Vector Machine achieves a precision of 0.88, whereas CNN achieves a greater precision of 98%. In contrast, with a smaller COREL1000 dataset, SVM achieves a precision of 86%, while CNN's precision is 83%. The inference drawn is that conventional machine learning excels with smaller datasets, whereas deep learning frameworks show improved recognition accuracy with huge samples of data.

Tawsifur Rahman, Muhammad E et al. proposed system was trained and tested using dataset chest X-ray pneumonia, wherein image data consists of five thousand two hundred and forty-seven chest X-ray pictures and this was later increased using data augmentation [10]. Three classification tasks are performed, Normal vs. Pneumonia: Distinguishing between normal X-rays and those showing signs of pneumonia. Bacterial vs. Viral Pneumonia: Identifying whether the pneumonia is caused by bacterial or viral infection. The outcomes demonstrated that DenseNet201 performs better than the three other unique deep CNN networks. It resulted with 98%, 93.3% and 95% of accuracy for the 3 classification schemes, respectively.

Table I: Summary table of the existing methodologies.

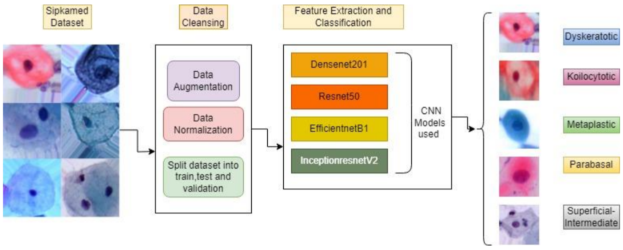
Reference	**Key Findings**	**Results**	**Limitations**
Wanli Liu et al. [1]	Hybrid approach with CNN and Visual Transformer for feature extractions, and Multilayer Perceptron for feature fusion. Tested on combined dataset of CRIC and SIPaKMeD.	Precision, accuracy and F1 of 91.8%, 91.7% and 91.7%.	Difficulty in distinguishing similar cell nuclei due to image ambiguity.
Anurag Tripathi et al. [2]	ResNet-152 used on SIPaKMeD dataset with transfer learning.	Classification accuracy of 94.9% achieved.	Need for dataset diversity.
Swati Shinde et al. [3]	Multiclass datasets converted into binary class (normal and abnormal). Workflow 1: PCA feature extraction with ensemble voting for classification. Workflow 2: Pre-trained with CNN with ANN for feature fusion and classification.	Accuracy of 97%, 99% and 100% for 7, 5 and 4-class classification.	Potential reduction in diagnostic accuracy due to class aggregation.
Wasswa William et al. [4]	Trainable Weka Segmentation; Simulated Annealing with Wrapper Filter feature selection; Fuzzy C-means classification.	A sensitivity of 99% and accuracy of 98% achieved.	Tested on small datasets. (917, 497, 60)
Alquran et al. [5]	New Feature Fusion method using Shuffle Net and Cervical Net. PCA used for feature extraction and dimensionality reduction.	Classification accuracy of 99.1% achieved.	Tested on Small dataset.(4049 images)

Taranjit Kaur et al. [6]	Use of pre-trained DCNN models(AlexNet, ResNet50, VGG16, GoogleNet, ResNet101, VGG19, InceptionV3, & InceptionResNetV2) with transfer learning for brain image classification.	AlexNet outperformed all other models.	Trained on small sample datasets.(50, 74, 160)
Pin Wang et al. [7]	Segmentation using Mean-shift clustering and classification based on nuclei features.	94.2% sensitivity and 93.4% specificity with 96% accuracy achieved.	Tested on small dataset.(362 images)
Pin Wang et al. [9]	Comparison of traditional machine learning(SVM) and deep learning(CNN) algorithm.	Machine learning performs better with small sample datasets and deep learning better with large datasets.	-
Mousser et al. [39]	Representation learning was done by transfer-learning techniques and categorisation of labels achieved through Multilayer-perceptron.	ResNet50 outperformed; Accuracy of 89.2% with five-fold cross-validation achieved.	Lower accuracy compared to expected standards.
Pal et al. [40]	Transfer learning models(backbone) with Deep metric learning(DML) for image feature discrimination; KNN classifier.	Contrastive, N-Pair embedding, and batch hard achieved mean-5 precision scores of 91.1%, 89.58%, and 85.14%, respectively.	Possible lack of generalizability during training. Class imbalance.
Md Mamunur Rahaman et al. [35]	Hybrid deep feature fusion(VGG16, VGG19, ResNet50, XceptionNet) technique for classification;	Accuracy of 99.1%(5-class) and 90.3%(7-class).	Performance degrades as class size increases.
Kalbhor et al. [41]	Discrete cosine transformer(DCT) and Haar transformer for feature extraction; Machine learning models for classification.	Classification accuracy of 81.1% achieved.	Lower accuracy; Low resolution images.
Chandana et al. [42]	SE-ResNet152 for classification; Deer Hunting Optimization for hyper-parameter tuning.	Precision, recall and F1-Score of 98.8%, 97.8% and 98.6%.	High dimensionality of concatenated features.
Li et al. [43]	Weekly supervised MHCRF based model for classification; Feature extraction of color, texture, and SOTA deep learning features.	Overall classification accuracy of 77.3% achieved.	Tested on small dataset(600 samples)

The related work section provides an extensive review of existing methodologies. The above summary table I compares the key findings, accuracy, and limitations of the reviewed studies.

3 Methodology

The following section includes the detailed explanation of the proposed methodology starting from pre-processing till classification.



Figureure 3.1: Proposed methodology framework.

Above Figureure 3.1 represents the framework of the proposed methodology. Each section of the framework is explained in detail as below.

3.1 Dataset description

To evaluate the proposed transfer learning methods, we utilized the Sipakmed Pap Smear dataset, a publicly available cervical cancer image dataset [11-14]. The Sipakmed dataset is a five-class dataset comprising a total of 4049 individual images featuring isolated cells were manually cropped from a collection of nine hundred and sixty-six cluster cell images. The cell classes are categorized into superficial-intermediate cells, parabasal cells, koilocytotic cells, Dysketarotic cells and Metaplastic cells. Dataset split ratio information is as follows: 80:5:15 for training, testing and validation respectively.

3.2 Data cleansing

The dataset's limited number of images were insufficient for effective model training, posing a risk of overfitting. We used data augmentation approaches to solve this issue, expanding the sample size using fundamental augmentation techniques. The transformation technique used are: rotation, height shift, wide shift, shear, zoom, horizontal flip, and fill mode. Table II below gives the detailed number of images class wise before and after augmentation.

Table II: Dataset description

Class no.	Class name	Number of original images	Number of images after augmentation
1	Dyskeratotic	813	1,602
2	Koilocytotic	825	1,610
3	Metaplastic	793	1,555
4	Parabasal	787	1,553
5	Superficial-Intermediate	831	1,630

Data augmentation techniques are basically used here to expand the amount and diversity of a sipkamed standard dataset as shown in Figure 3.2, hence improving the performance and robustness of CNN models. Geometric tranformations utilized here are: Rotation with a degree of 20, width_shift_range, height_shift_range, shear_range, zoom_range equal to 0.2, Flipping of images is done horizontally to help the model learn to recognize images from different perspectives and reduce the generalization errors which can be witnessed in the section 4.2.

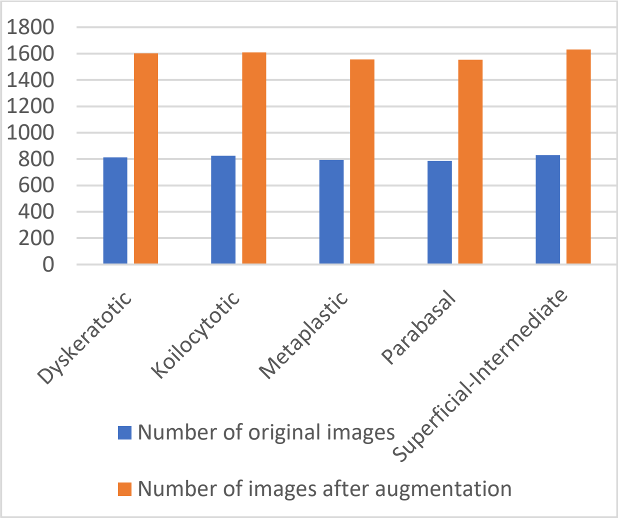


Figure 3.2 Bar graph representing number of images before and after augmentation.

3.3 Representation learning and classification

Convolutional neural networks (CNN) are artificial neural networks that specialize in detecting and understanding patterns, making them helpful for image analysis [15-19]. CNN consists of multiple layers, each with its own set of functions to perform on the input data. The architecture of the convolutional neural network comprises various layers, including convolutional, Rectified Linear Unit (ReLu) [39], and down-sampling layers with max strategy.

The design is influenced by AlexNet, as outlined in the next section and onward. It comprises of six layers: Conv-2D, Re-Lu, down-sampling, and a dense layer. The input layer stores raw data, the convolutional layer calculates the dot-product of patches of imagery and kernel filters, and an activation function is applied by the activation function layer to each portion of the output of the convolutional layer.

The layer of down sampling improves the efficiency of memory output from the preceding layers, resulting in decreased computational costs. The Dense layer receives input from its preceding layer and produces the calculated one-dimensional array of logit values. Techniques for detecting and objects' racking involve extracting features from images and videos, primarily applied for security purposes.

Enhancing training performance can be achieved by incorporating extra layers, like dropouts. The layer called drop-out is specifically initiated during the period of training. Specifically, in the forward pass (input to the function). The dropout layer selectively eliminates a specified number of neurons and retains those that persist. Only the non-dropped neurons are updated during the backward pass. This dropout mechanism serves as a regularization technique. By allowing the model to acquire robust, neuron-independent characteristics, it prevents the learning of features that are overly dependent on specific neurons, thus mitigating over fitting during the training phase.

CNN, or Convolutional Neural Networks [20], has a significant impact in classification of images. Its approach is based on the use of numerous tiny kernels to determine the characteristics in every layer. CNN is made up of a collection of layers, which can extract various features from the input images. Deep feature extraction from SIPaKMeD datasets ‘of various classes images of original size 224x224x3 were accomplished by extracting features from pre-trained CNN models. Resnet50, Densenet201, EfficientnetB1, InceptionresnetV2 are the various CNN architectures used in this article.

Base ResNet50, Densenet201, Efficientnetb2, Inceptionnetv2 models are customised as described in the following sections. Hyper-parameters used in each of these architectures are detailed as follows: Standard sipkamed dataset is splitted into the training instances ratio of 80% and validation instances of 20%. Learning rate of 0.001 and adam optimizer are used for reducing the loss during training, dropout rate of 20% and L2 regularizers are utilized to overcome over fitting. Around 100 iterations were used to scan the entire dataset with the batch size of 32 to ease the training process. By choosing these hyper-parameters values, we have achieved good results compared to the existing work.

3.3.1 Resnet50

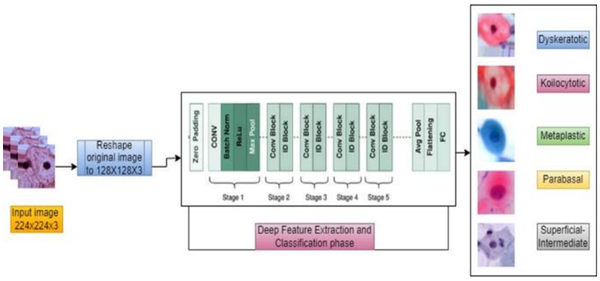


Figure 3.3 Base Resnet50 architecture.

ResNet-50 typically takes an input image of size 224x224 pixels. A standard convolutional layer with 7x7 filters and 64 output channels. Followed by batch normalization and ReLU activation. The core building blocks of ResNet are residual blocks [16]. ResNet-50 consists of 16 residual blocks grouped into four stages. Each stage contains a different number of residual blocks with varying numbers of filters. The ResNet50 [21-24] architecture comprises 50 trainable layers, with 48 of them being convolutional layers, in addition to a solitary layer for both average pooling and max-pooling. Spatial pooling is applied at different stages of the network to reduce spatial dimensions. Global Average Pooling (GAP) is used before making the layer dense. The final network layer incorporates a fully-connected layer that employs softmax activation to classify five classes in this context.

The number of nodes in the last layer aligns with the number of classes involved in the classification task. Diverging from its forerunners, ResNet50's shortcut connections deviate from the usual two layers and instead bypass three layers. The Figureure 3.3 illustrates the ResNet50 architecture, highlighting the point Res-Nets support two types of shortcut- connections. If the dimensions of both inputs and outputs are similar [25-28], straight-arrow connections are utilized. In cases where there is an increase in these dimensions, the alternative connections marked with dotted lines are activated. Parameter details of Resnet50 architecture are listed below in Table III.

Table III: Parameter details of Resnet50 architecture.

Type of the layer	Input shape	Output shape	Number of parameters
Image	224 x 224 x 3	128 x 128 x 3	0
Base model	128 x 128 x 3	4 x 4 x 2048	23,534,592
Global average pooling	4 x 4 x 2048	2048	0
Dense	2048	1024	2,098,176
Dropout	1024	1024	0
Dense (output layer)	1024	5	5,125
Total parameters			25,637,893

3.3.2 Densenet201

Initial input image size, convolution layers and their filter sizes in Densenet201 are similar to Resnet50. DenseNet concatenates feature maps from previous layers, propagating them to subsequent layers and connecting them to newly created feature maps. DenseNet offers benefits such as feature reuse and less issues with exploding or vanishing gradients [29].

The structure of DenseNet is composed of an input layer, three dense blocks, layers of transition, and a down-sampling with average strategy globally. Layers of the transition consist of a batch-normalization layer, a convolution layer of size 1×1, and a window size of 2×2 average down-sampling layer including step size of 2 as shown in Figure 3.3. Unlike conventional pooling algorithms, global average pooling (GAP) reduces a feature map from w × w × c to 1 × 1 × c, effectively condensing the entire slice into a single digit. The final dense-layer, is modified due to the original design of the last FCL to identify five categories [30-31]. Parameter details of Densenet201 architecture are listed below in Table IV.

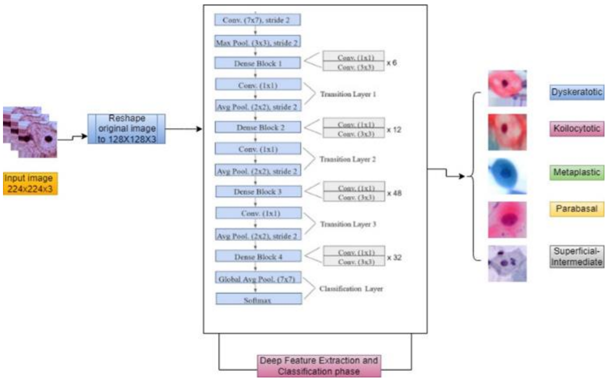


Figure 3.4 Base Densenet201 architecture.

Table IV: Parameter details of Densenet201 architecture.

Type of the layer	Input shape	Output shape	Number of parameters
image	224 x 224 x 3	128 x 128 x 3	0
Base model	128 x 128 x 3	4 x 4 x 1920	18,092,928
Global average pooling	4 x 4 x 1920	1920	0
Dense	1920	1024	1,967,104
Dropout	1024	1024	0
Dense (output layer)	1024	5	5,125
Total parameters			20,065,157

3.3.3 EfficientnetB1

EfficientNetB1 belongs to the EfficientNet series of convolutional neural networks (CNNs) formulated to achieve superior accuracy and efficiency in contrast to conventional CNN designs. The architecture of EfficientNetB1 is derived from a compound scaling approach, which uniformly adjusts the network's width, depth, and resolution. [32]. This allows the model to be more efficient in terms of computational resources while maintaining high accuracy. The "B1" in EfficientNetB1 denotes the specific scaling coefficients used for the width, depth, and resolution. The Efficientnet-B1 architecture's up-sampling network is made up of decoder blocks. Each block consists of up-sampling of window size 2x2 convolution 2D of the output from the previous layer, with a step size of two. This result is then concatenated [32] with the representation learning maps from the section of the encoder. Before proceeding to the next decoder [32] block, the combined tensor is passed through two convolution layers with Re-LU activation and Batch- Normalization as shown in Figure 3.4. The architecture's last layer is a soft-max convolution with the same number of channels as the output classes, which is five, and the output image size is similar as the input. Parameter details of EfficientnetB1 architecture are listed below in Table V.

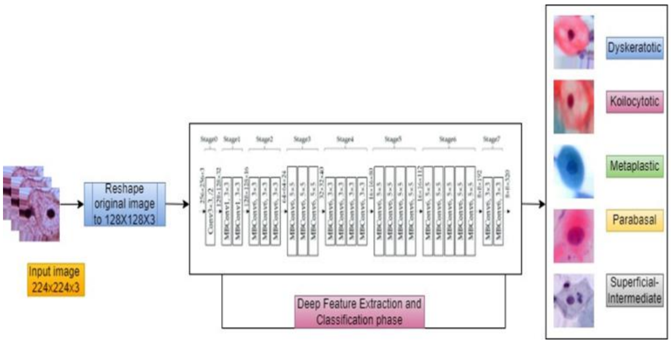


Figure 3.5 Base EfficientnetB1 Architecture [18].

Table V: Parameter details of EfficientnetB1 architecture

Type of the layer	Input shape	Output shape	Number of parameters
image	224 x 224 x 3	128 x 128 x 3	0
Base model	128 x 128 x 3	4 x 4 x 1280	6,513,184
Global average pooling	4 x 4 x 1280	1280	0
Dense	1280	1024	1,311,744
Dropout	1024	1024	0
Dense (output layer)	1024	5	5,125
Total parameters			7,830,053

3.3.4 InceptionresnetV2

The Inception-ResNet-v2 model merges the Inception structure with Residual connections. The fundamental characteristics [33] of the Inception-ResNet-v2 architecture include:

Inception Blocks (Inception modules): The network employs Inception blocks, comprising several parallel convolutional branches with varying kernel sizes. These branches capture features at various scales. Inception modules help the network learn diverse and rich representations. Residual Connections: In addition to Inception modules, Inception-ResNet-v2 incorporates residual connections. Residual connections aid in addressing the vanishing gradient problem and enable the training of extremely deep networks. These connections involve adding the input of a layer to its output, creating a shortcut connection. Reduction Blocks: Similar to the original Inception architecture, Inception-ResNet-v2 includes reduction blocks that simultaneously reduce the spatial dimensions of the input attribute maps and increase the number of channels. Top of Form This helps in reducing computational complexity and extracting high-level features. Stem Block: Inception-ResNet-v2 has a stem block at the beginning of the network, which processes the input image and extracts initial features. The stem block typically includes a blend of layers of pooling and convolution, as demonstrated in Figureure 3.5. Auxiliary Classifiers: Inception-ResNet-v2 includes auxiliary classifiers during training to provide additional gradients for the earlier layers. This helps with the training of very deep networks. The Inception-Resnet [31] block combines different sized convolutional filters using residual connections. Using residual connections prevents degradation from deep structures and reduces training time. Parameter details of InceptionResnetV2 architecture are listed below in Table VI.

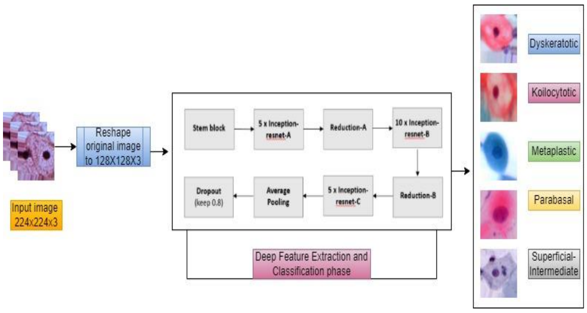


Figure 3.6 Base InceptionResnetV2 architecture.

Table VI: Parameter details of InceptionResnetV2 architecture.

Type of the layer	Input shape	Output shape	Number of parameters
image	224 x 224 x 3	128 x 128 x 3	0
Base model	128 x 128 x 3	2 x 2 x 1536	54,276,192
Global average pooling	2 x 2x 1536	1536	0
Dense	1536	1024	1,573,888
Dropout	1024	1024	0
Dense (output layer)	1024	5	5,125
Total parameters			55,855,205

4 Results and discussions

The Sipakmed dataset consists of five classes and includes 4049 individual images of isolated cells, manually extracted from 966 clustered cell images. The cell classes are categorized into superficial-intermediate cells, parabasal cells, koilocytotic cells, Dysketarotic cells and Metaplastic cells. Partitioning the SIPaKMeD dataset involves allocating 80% for training, 5% for testing, and the remaining 15% for validation. Following evaluation metrics have been used in the article for the categorization of Cervix cancer cells:

Accuracy, defined as the percentage of successfully forecasted rows to total rows based on the five categories supplied in the dataset. Precision determines how many anticipated outcomes are actually the same class label. Recall assesses the model's capacity to accurately identify all relevant instances within a dataset; F1-score is the balance of precision and recall, offering a single measure that optimizes the model's accuracy and recall. These metrics are calculated using the subsequent formulas as specified from equations 4.1 through 4.4:

$$\text{Accuracy}=\frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})} \qquad \text{--------}\rightarrow \quad 4.1$$

$$\text{Precision}=\frac{TP}{TP+FP} \qquad \text{---------}\rightarrow \quad 4.2$$

$$\text{Recall}=\frac{TP}{TP+FN} \qquad \text{---------}\rightarrow \quad 4.3$$

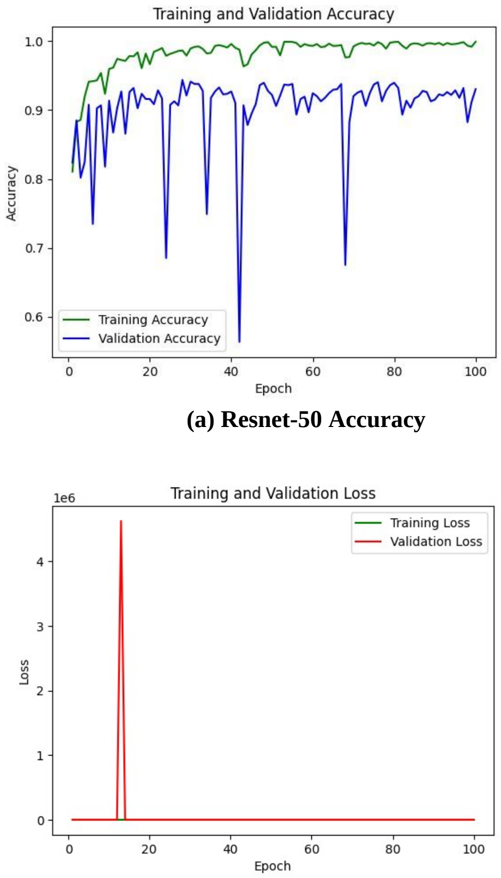
$$\text{F1-Score}=\frac{2*Precision*Recall}{Precision+Recall} \qquad \text{---------}\rightarrow \quad 4.4$$

While accuracy is a common metric, it may not be sufficient in cases of imbalanced datasets. As SIPaKMeD dataset is balanced, accuracy can be considered as the most relevant metric in our research. The Error Matrix is a tabular representation that presents the numbers of correctly classified cancer cells and misclassified cancerous cells. This helps understand the types of errors made by the model.

4.1 SIPaKMeD dataset classification before augmentation

Figure 4.1, 4.3 4.5 and 4.7 shows the accuracy and loss graphs of four CNN architectures plotted against training and validation data. Comparing training and validation accuracy helps assess whether the system generalizes well to new data or if there's over fitting, which is overcome by adding L2 regularisers in all the four models. The Loss graph represents the decrease in the model's loss on the training set over 100 epochs. A decreasing training loss indicates that the model is learning and adjusting its weights to minimize the error as and when epochs are increased. Monitoring validation loss helps identify overfitting. If the training loss keeps decreasing while the validation loss rises, it indicates that the model may be memorizing the training data and demonstrating suboptimal performance when presented with new, unseen data. Hence, along with the training dataset, model's loss and accuracy on a separate validation set are also depicted. Among Resnet50, Densenet201, Efficientnetb1 and InceptionResnetv2 models, Densenet201 achieved the highest accuracy of 96.74.

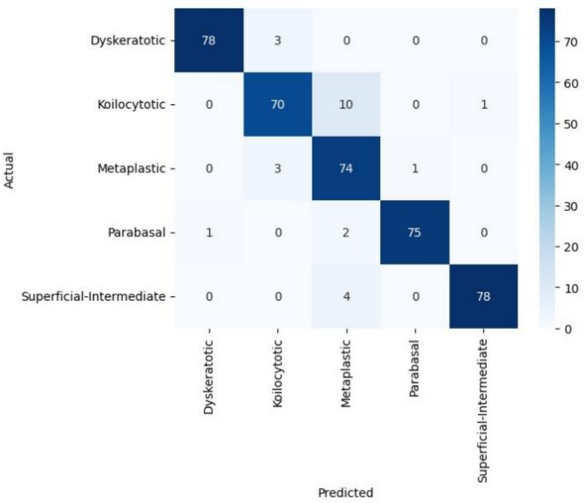
The Confusion matrix of the proposed methodology is represented in Figure 4.2,4.4,4.6 and 4.8. It offers an overview of the forecasts of the network in contrast to the real ground truth for five distinct classes. The matrix is particularly useful for understanding the types and frequencies of errors these four CNN models make [34].



(a) Resnet-50 Accuracy

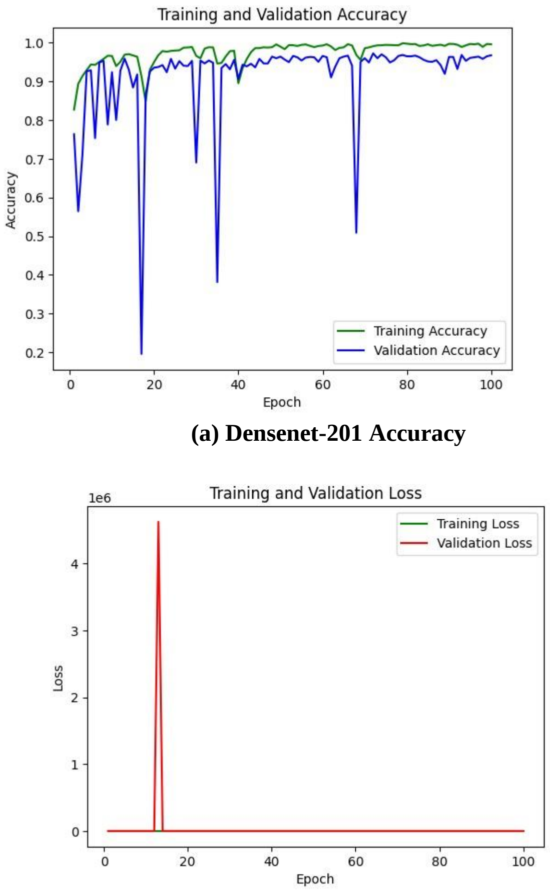
(b) Resnet-50 Loss

Figure 4.1 Accuracy (a) and Loss (b) graphs of Training and Validation datasets of SIPaKMeD images for Resnet50 architecture.



	Dyskeratotic	Koilocytotic	Metaplastic	Parabasal	Superficial-Intermediate
Dyskeratotic	78	3	0	0	0
Koilocytotic	0	70	10	0	1
Metaplastic	0	3	74	1	0
Parabasal	1	0	2	75	0
Superficial-Intermediate	0	0	4	0	78

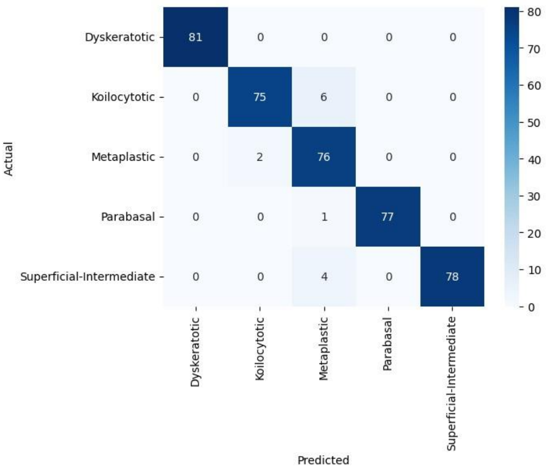
Figure 4.2 Confusion matrix obtained for Resnet50.



(a) Densenet-201 Accuracy

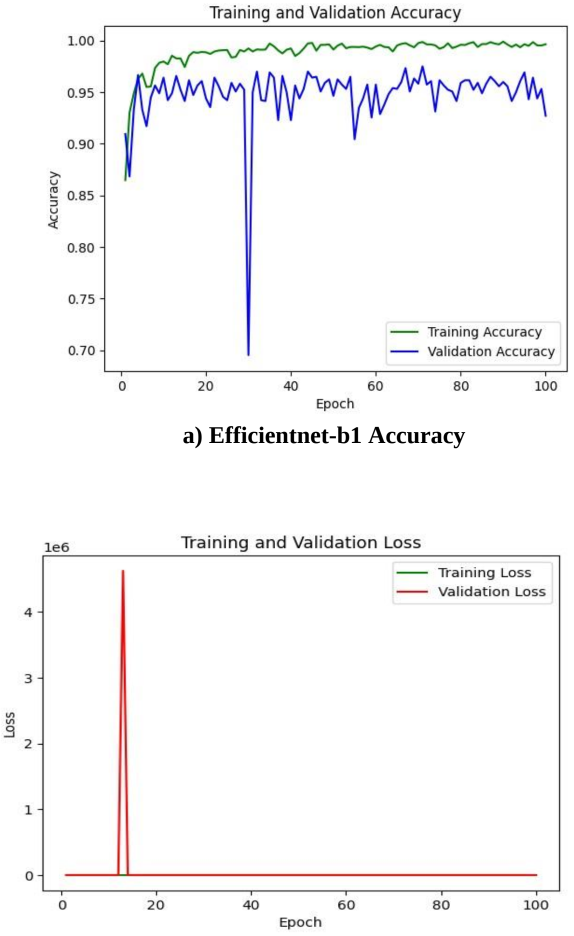
(b) Densenet-201 Loss

Figure 4.3 Accuracy (a) and Loss (b) graphs of Training and Validation datasets of SIPaKMeD images for Densenet201 architecture



	Dyskeratotic	Koilocytotic	Metaplastic	Parabasal	Superficial-Intermediate
Dyskeratotic	81	0	0	0	0
Koilocytotic	0	75	6	0	0
Metaplastic	0	2	76	0	0
Parabasal	0	0	1	77	0
Superficial-Intermediate	0	0	4	0	78

Figure 4.4 Confusion matrix obtained for Densenet201



a) Efficientnet-b1 Accuracy

(b) Efficientnet-b1 Loss

Figure 4.5 Accuracy (a) and Loss (b) graphs of Training and Validation datasets of SIPaKMeD images for EfficientnetB1 architecture

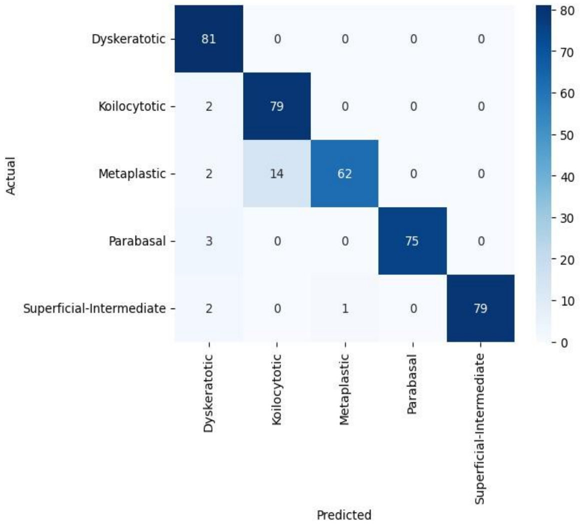
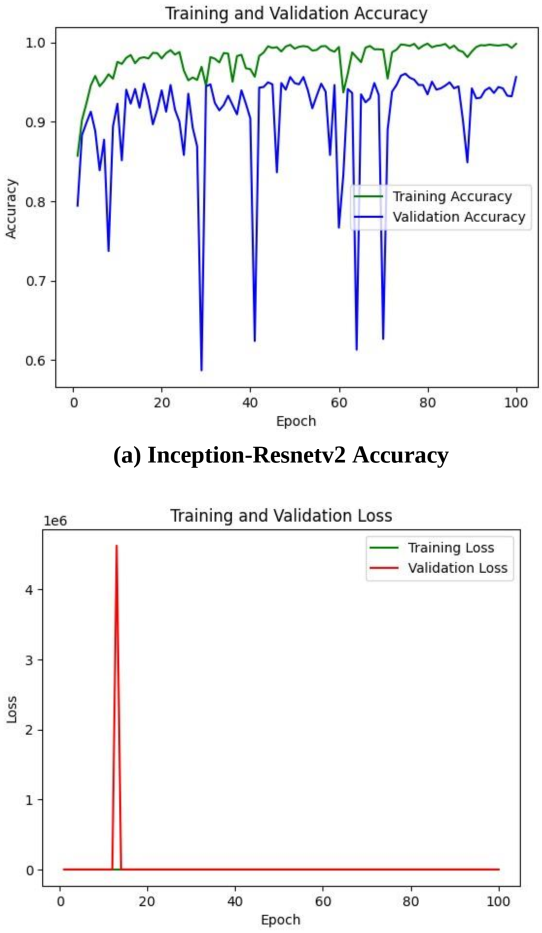


Figure 4.6 Confusion matrix obtained for EfficientnetB1



(a) Inception-Resnetv2 Accuracy

(b) Inception-Resnetv2 Loss

Figure 4.7 Accuracy (a) and Loss (b) graphs of Training and Validation datasets of SIPaKMeD images for InceptionResnetV2 architecture.

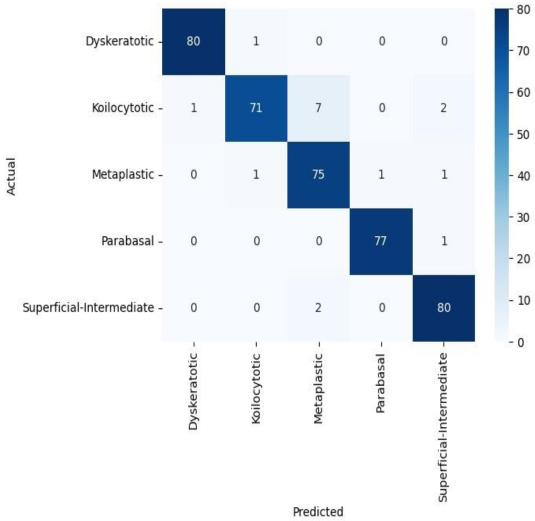


Figure 4.8 Confusion matrix obtained for InceptionResnetV2

4.2 SIPaKMeD dataset classification after augmentation

Different techniques for augmenting data including rotation, height shift, width shift, shear, zoom, horizontal flipping, and fill mode are used on SIPaKMeD dataset which increased number of images from 4049 to 7950. This has increased accuracy as described below. An improved accuracy and reduction in the loss of all the CNN models after augmentation are described in the Figureures 4.10, 4.12, 4.14 and 4.16.

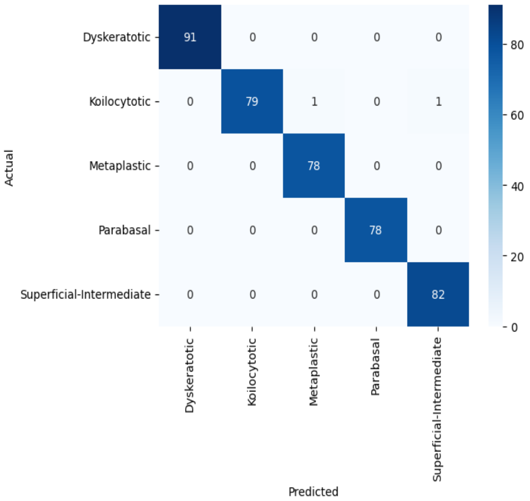
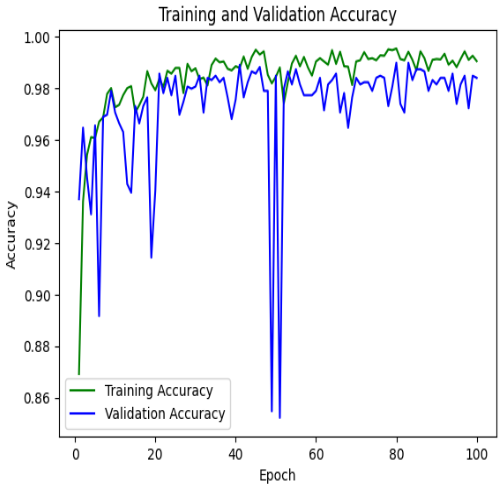
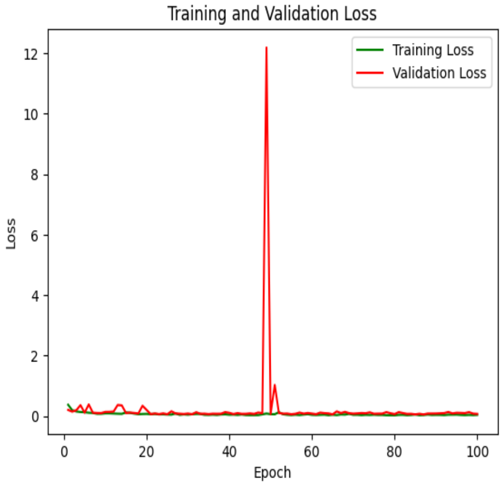


Figure 4.9 Confusion matrix obtained for Resnet50.



(a) Resnet-50 Accuracy



(b) Resnet-50 Loss

Figure 4.10 Accuracy (a) and Loss (b) graphs of Training and Validation datasets of SIPaKMeD images for Resnet50 architecture.

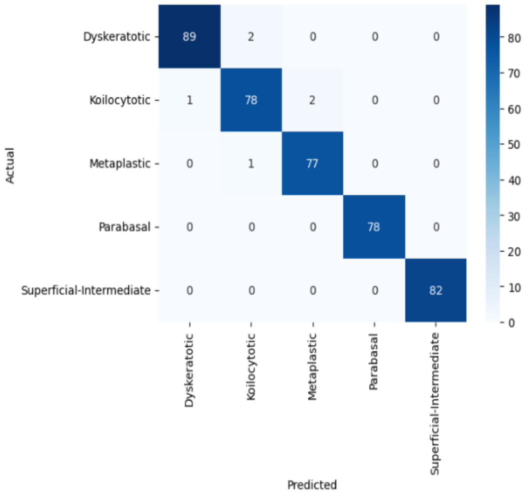
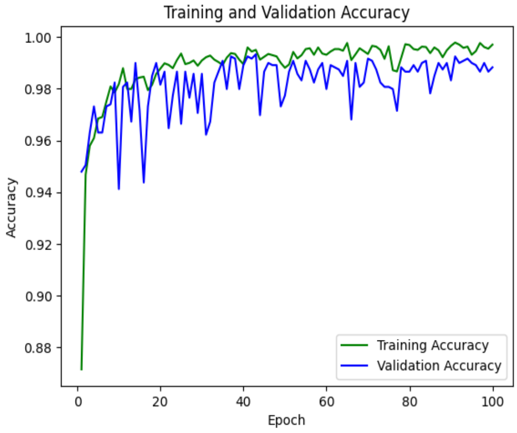
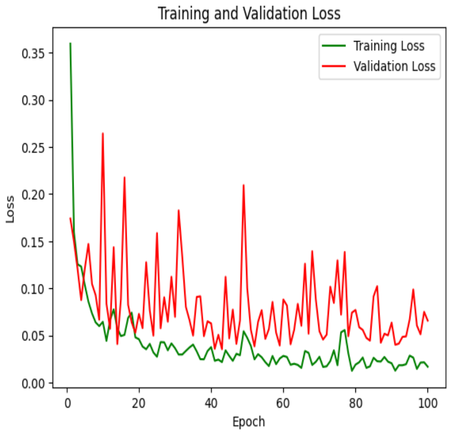


Figure 4.11 Confusion matrix obtained for Densenet201.



(a) Densenet-201 Accuracy



(b) Densenet-201 Loss

Figure 4.12 Accuracy and Loss graphs of Training and Validation datasets of SIPaKMeD images for Densenet201 architecture.

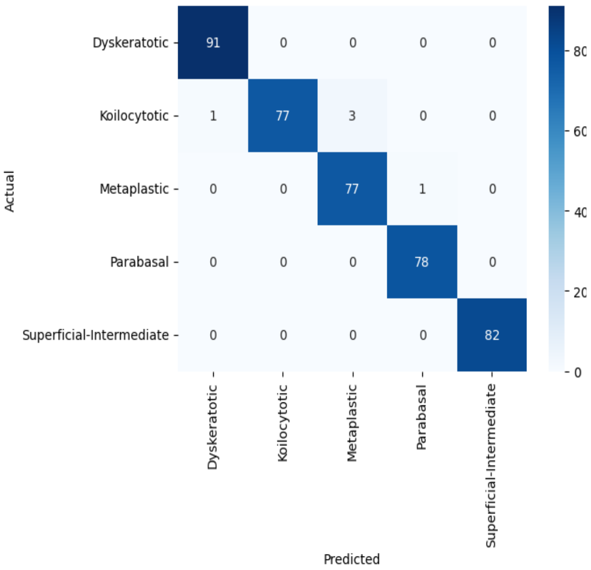
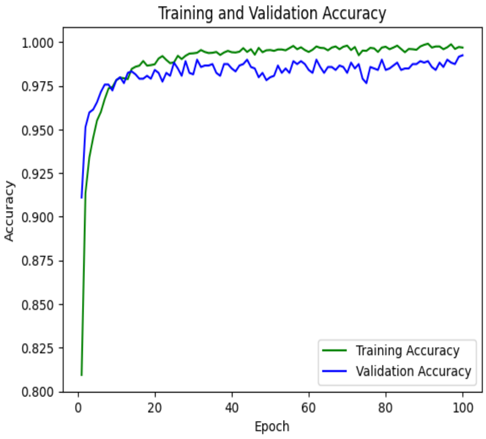
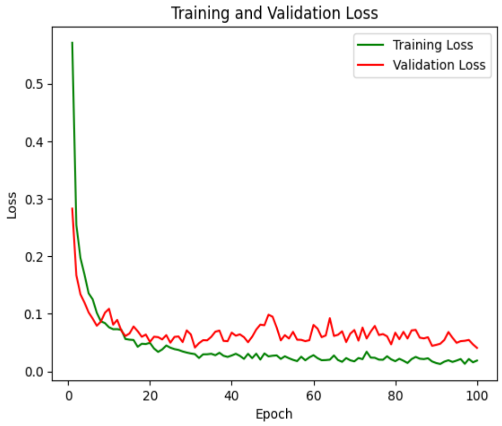


Figure 4.13 Confusion matrix obtained for Efficientnetb1.



(a) Efficientnet-b1 Accuracy



(b) Efficientnet-b1 Loss

Figure 4.14 Accuracy and Loss graphs of Training and Validation datasets of SIPaKMeD images for Efficientnetb1 architecture.

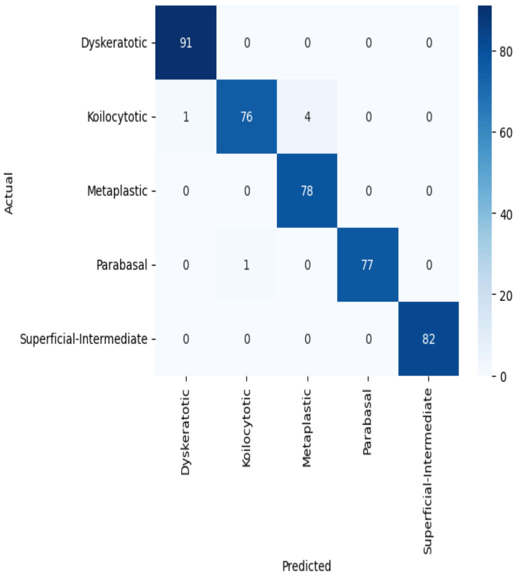
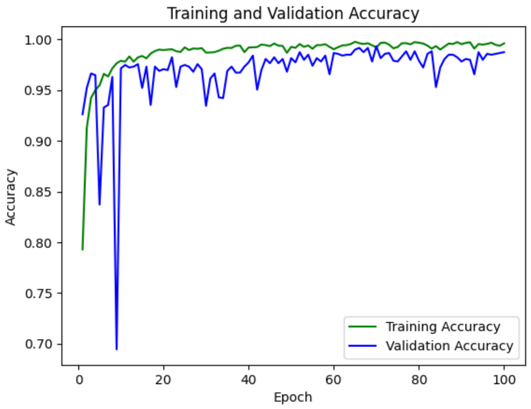
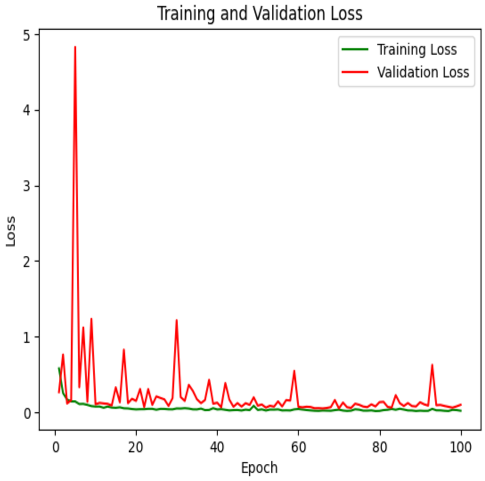


Figure 4.15 Confusion matrix obtained for InceptionResnetv2

The confusion matrix after the augmentation are shown in the Figure 4.9, 4.11,4.13 and 4.15.



(a) Inception-Resnetv2 Accuracy



(b) Inception-Resnetv2Loss

Figure 4.16 Accuracy and Loss graphs of Training and Validation datasets of SIPaKMeD images for InceptionResnetv2 architecture.

Table VII: Performance comparison between existing and proposed models.

Reference	Model	Accuracy	Precision
[35]	VGG16 VGG19 XceptionNet	98.27% 96.43% 65.77%	0.98 0.96 0.75
[36]	Bagging ensemble classifier	94.09%	
[37]	DenseNet-161	98.96%	
[3]	Voting ensemble (RF, SVM, FCNN)	88.95%	0.99
[2]	Resnet- 152	94.89%	
Proposed models	EfficientNet -b1 InceptionNet-Resnet-v2 DenseNet-201 Resnet -50	98.78% 98.53% 98.53% 99.51%	0.99 0.99 0.99 1.00

CNN architectures used in this work are compared against other ML and DL techniques as stated in table VII. The detailed metrics evaluation for the following proposed CNN [38] models are listed in the below table VIII. Resnet50 outperformed compared to the other three models.

Table VIII: Performance comparison of proposed models.

Sl No	CNN model	Class labels	Accuracy	Precision	Recall	F1-Score
1	Resnet -50	Dysketarotic	1.00	1.00	1.00	1.00
		koilocytotic	1.00	1.00	0.98	0.99
		Metaplastic	1.00	0.99	1.00	0.99
		parabasal	1.00	1.00	1.00	1.00
		superficial-intermediate	1.00	0.99	1.00	0.99
2	InceptionNet-Resnet-v2	Dysketarotic	0.99	0.99	1.00	0.99
		koilocytotic	0.99	0.99	0.94	0.96
		Metaplastic	0.99	0.95	1.00	0.97
		parabasal	0.99	1.00	0.99	0.99
		superficial-intermediate	0.99	1.00	1.00	1.00
3	EfficientNet -b1	Dysketarotic	0.99	0.99	1.00	0.99
		koilocytotic	0.99	1.00	0.95	0.97
		Metaplastic	0.99	0.96	0.99	0.97
		parabasal	0.99	0.99	1.00	0.99
		superficial-intermediate	0.99	1.00	1.00	1.00
4	DenseNet-201	Dysketarotic	0.99	0.99	0.98	0.98
		koilocytotic	0.99	0.96	0.96	0.96
		Metaplastic	0.99	0.97	0.99	0.98
		parabasal	0.99	1.00	1.00	1.00
		superficial-intermediate	0.99	1.00	1.00	1.00

5 Conclusion

This paper presents the multiclass classification of cervical cancer dataset named SIPaKMeD of five different classes using various CNN architectures namely Resnet50, Densenet201, Efficientnetb1, Inceptionresnetv2 and obtained an average accuracy of 95% on test data. Among these techniques, Densenet201 outperformed with the highest accuracy of 96.74%. After performing data augmentation with various techniques on the SIPaKMeD dataset, average accuracy has been increased to 98.78% with a great reduction in the loss and misclassification. Resnet50 won the race with the highest accuracy of 99.5% after augmentation. Moreover, additional datasets can be incorporated for conducting classification analysis of cervical cancer. Additionally, deep learning architectures can facilitate stage-wise predictions in this context.

References

[1] Wanli Liu, Chen Li, Ning Xu, Tao Jiang, Md Mamunur Rahaman, Hongzan Sun, Xiangchen Wu, Weiming Hu, Haoyuan Chen, Changhao Sun, Yudong Yao, Marcin Grzegorzek, “CVM-Cervix: A hybrid cervical Pp-smear image classification framework using CNN, visual transformer and multilayer perceptron” – 2022. https://doi.org/10.1016/j.patcog.2022.108829

[2] Anurag Tripathi, Aditya Arora, Anupama Bhan, “Classification of cervical cancer using Deep Learning Algorithm” -2021. http://dx.doi.org/10.1109/ICICT50816.2021.9358570

[3] Swati Shinde, Madhura Kalbhor, Pankaj Wajire, “DeepCyto: a hybrid framework for cervical cancer classification by using deep feature fusion of cytology images” – 2022. http://dx.doi.org/10.3934/mbe.2022301

[4] Wasswa William, Andrew Ware, Annabella Habinka Basaza-Ejiri, Johnes Obungoloch, “A pap-smear analysis tool (PAT) for detection of cervical cancer from pap-smear images” – 2019. https://doi.org/10.1186/s12938-019-0634-5

[5] Alquran, H.; Alsalatie, M.; Mustafa, W.A.; Abdi, R.A.; Ismail, A.R. “Cervical Net: A Novel Cervical Cancer Classification Using Feature Fusion”- 2022. https://doi.org/10.3390%2Fbioengineering9100578

[6] Taranjit Kaur, Tapan Kumar Gandh, “Deep convolutional neural networks with transfer learning for automated brain image classification” – 2020. https://doi.org/10.1007/s00138-020-01069-2

[7] Pin Wang, Lirui Wang, Yongming Li, Qi Song, Shanshan Lv, Xianling Hu, “Automatic cell nuclei segmentation and classification of cervical Pap smear images” – 2018. https://doi.org/10.1016/j.bspc.2018.09.008

[8] Md Mamunur Rahaman, Chen Li, Xiangchen Wu, Yudong Yao, Zhijie Hu, Tao Jiang, Xiaoyan Li, Shouliang Qi, “A Survey for Cervical Cytopathology Image Analysis Using Deep Learning” – 2020. https://doi.org/10.1109/ACCESS.2020.2983186

[9] Pin Wang, En Fan, Peng Wang, “Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning” – 2020. https://doi.org/10.1016/j.patrec.2020.07.042

[10] Tawsifur Rahman, Muhammad E. H. Chowdhury, Amith Khandakar, Khandaker R. Islam, Khandaker F. Islam, Zaid B. Mahbub, Muhammad A. Kadir, Saad Kashem, “Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray” – 2020. http://dx.doi.org/10.3390/app10093233

[11] Kong, Shien Nie, et al. "Cervical Cancer Identification using Deep Learning Approaches." 2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED). IEEE, 2022. https://doi.org/10.1109/ICCED56140.2022.10010544.

[12] Gorantla, Rohan, et al. "Cervical cancer diagnosis using cervixnet-a deep learning approach." 2019 IEEE 19th international conference on bioinformatics and bioengineering (BIBE). IEEE, 2019. https://doi.org/10.1109/BIBE.2019.00078

[13] Hemalatha, K., and V. Vetriselvi. "Deep Learning based Classification of Cervical Cancer using Transfer Learning." 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC). IEEE,2022. https://doi.org/10.1109/ICESIC53714.2022.9783560

[14] Behnam Neyshabhr, Hanie Sedghi, Chiyun Zhang, “What is being transferred in transfer learning?” – 2020. https://doi.org/10.48550/arXiv.2008.11687

[15] Nanni, L., Ghidoni, S., Brahnam, S., “Ensemble of convolutional neural networks for bioimage classification”2021. https://doi.org/10.1016/j.aci.2018.06.002

[16] https://www.educba.com/keras-resnet50/.

[17] Jaiswal, Aayush, Neha Gianchandani, Dilbag Singh, Vijay Kumar, and Manjit Kaur. "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning." Journal of Biomolecular Structure and Dynamics 39, no. 15 (2021): 5682-5689. http://dx.doi.org/10.1080/07391102.2020.1788642

[18] Jie, Yongshi, Xianhua Ji, Anzhi Yue, Jingbo Chen, Yupeng Deng, Jing Chen, and Yi Zhang. "Combined multi-layer feature fusion and edge detection method for distributed photovoltaic power station identification." Energies 13, no. 24 (2020): 6742. https://doi.org/10.3390/en13246742

[19] Krishna Adithya, Venkatesh, Bryan M. Williams, Silvester Czanner, Srinivasan Kavitha, David S. Friedman, Colin E. Willoughby, Rengaraj Venkatesh, and Gabriela Czanner. "EffUnet-SpaGen: an efficient and spatial generative approach to glaucoma detection." Journal of Imaging 7, no. 6 (2021): 92. http://dx.doi.org/10.3390/jimaging7060092

[20] Nguyen, Long D., Dongyun Lin, Zhiping Lin, and Jiuwen Cao. "Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation." In 2018 IEEE international symposium on circuits and systems (ISCAS), pp. 1-5. IEEE, 2018. https://doi.org/10.1109/ISCAS.2018.8351550

[21] Salama, Gerges M., Asmaa Mohamed, and Mahmoud Khaled Abd-Ellah. "COVID-19 classification based on a deep learning and

machine learning fusion technique using chest CT images." Neural Computing and Applications (2023):1-19. http://dx.doi.org/10.1007/s00521-023-09346-7.

[22] Jogin, Manjunath, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva. "Feature extraction using convolution neural networks (CNN) and deep learning." In 2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT), pp. 2319-2323. IEEE, 2018. https://doi.org/10.1109/RTEICT42901.2018.9012507.

[23] M. E. Plissiti, P. Dimitrakopoulos, G. Sfikas, C. Nikou, O. Krikoni, A. Charchanti, Sipakmed: A new dataset for feature and image based classification of normal and pathological cervical cells in Pap smear images, in IEEE International Conference on Image Processing (ICIP) 2018, (2018), 3144–3148. https://doi.org/10.1109/ICIP.2018.8451588

[24] SIPakMed Database, 2022. Available from: https://www.cs.uoi.gr/~marina/sipakmed.html.

[25] Guo lili, Ding shifei, "Deep learning research progress [J]", 2015.

[26] Akshay Mangawati, Mohana, Mohammed Leesan, H. V. Ravish Aradhya, "Object Tracking Algorithms for video survilllance applications" International conference on communication and signal processing (ICCSP), India, 2018, pp. 0676-0680. https://doi.org/10.1109/ICCSP.2018.8524260

[27] Apoorva Raghunandan, Mohana, Pakala Raghav and H. V. Ravish Aradhya, "Object Detection Algorithms for video survilllance applications" International conference on communication and signal processing (ICCSP), India, 2018, pp. 0570-0575. https://doi.org/10.1109/ICCSP.2018.8524461

[28] Kibriya, Hareem, and Rashid Amin. "A residual network-based framework for COVID-19 detection from CXR images." Neural Computing and Applications 35, no. 11 (2023): 8505-8516. https://doi.org/10.1007/s00521-022-08127-y

[29] Wang, Shui-Hua, and Yu-Dong Zhang. "DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16, no. 2s (2020): 1-19. https://doi.org/10.1145/3341095

[30] Krishna Adithya, Venkatesh, Bryan M. Williams, Silvester Czanner, Srinivasan Kavitha, David S. Friedman, Colin E. Willoughby, Rengaraj Venkatesh, and Gabriela Czanner. "EffUnet-SpaGen: an efficient and spatial generative approach to glaucoma detection." Journal of Imaging 7, no. 6 (2021): 92. https://doi.org/10.3390/jimaging7060092

[31] Nguyen, Long D., Dongyun Lin, Zhiping Lin, and Jiuwen Cao. "Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation." In 2018 IEEE international symposium on circuits and systems (ISCAS), pp. 1-5. IEEE, 2018. https://doi.org/10.1109/ISCAS.2018.8351550

[32] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International conference on machine learning, pp. 6105-6114. PMLR, 2019. https://doi.org/10.48550/arXiv.1905.11946.

[33] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1. 2017. http://dx.doi.org/10.1609/aaai.v31i1.11231

[34] Madhusudhan, M. V., V. Udayarani, and C. Hegde. "Finger vein based authentication using deep learning techniques." Int. J. Recent Technol. Eng 8, no. 5 (2020): 5403-5408. http://dx.doi.org/10.1088/1757-899X/1105/1/012032

[35] Md Mamunur Rahaman, Chen Li, Yudong Yao, Frank Kulwa, Xiangchen Wu, Xiaoyan Li, Qian Wang, "DeepCervix: A deep learning-based framework for the classification of cervical cells using hybrid deep feature fusion techniques" – 2021. https://doi.org/10.1016/j.compbiomed.2021.104649

[36] Kyi Pyar Win, Yuttana Kitjaidure, Kazuhiko Hamamoto, Thet Myo Aung, "Computer-Assisted Screening for Cervical Cancer Using Digital Image Processing of Pap Smear Images" – 2020.

[37] Muhammed Talo, "Diagnostic Classification of Cervical Cell Images from Pap Smear Slides"- 2019. http://dx.doi.org/10.33793/acperpro.02.03.116

[38] Shruthi, U., and V. Nagaveni. "TomSevNet: a hybrid CNN model for accurate tomato disease identification with severity level assessment." Neural Computing and Applications 36, no. 10 (2024): 5165-5181. http://dx.doi.org/10.1007/s00521-023-09351-w

[39] Mousser, Wafa, and Salima Ouadfel. "Deep feature extraction for pap-smear image classification: A comparative study." In Proceedings of the 2019 5th International Conference on Computer and Technology Applications, pp. 6-10. 2019. http://dx.doi.org/10.1145/3323933.3324060

[40] Pal, Anabik, Zhiyun Xue, Brian Befano, Ana Cecilia Rodriguez, L. Rodney Long, Mark Schiffman, and Sameer Antani. "Deep metric learning for cervical image classification." IEEE Access 9 (2021): 53266-53275. https://doi.org/10.1109/ACCESS.2021.3069346

[41] Kalbhor, Madhura, Swati Vijay Shinde, and Hemant Jude. "Cervical cancer diagnosis based on cytology pap smear image classification using fractional coefficient and machine learning classifiers." TELKOMNIKA (Telecommunication Computing Electronics and Control) 20, no. 5 (2022): 1091-1102. http://doi.org/10.12928/telkomnika.v20i5.22440

[42] Chandana, B. Sai. "Multi-class Cervical Cancer Classification using Transfer Learning-based Optimized SE-ResNet152 model in Pap Smear Whole Slide Images." International journal of electrical and computer engineering systems 14, no. 6 (2023): 613-623. http://dx.doi.org/10.32985/ijeces.14.6.1

[43] Li, Chen, Hao Chen, Le Zhang, Ning Xu, Dan Xue, Zhijie Hu, He Ma, and Hongzan Sun. "Cervical histopathology image classification using multilayer hidden conditional random fields and weakly supervised learning." Ieee Access 7 (2019): 90378-90397. https://doi.org/10.1109/ACCESS.2019.2924467

Animation Character Mouth Matching Model Considering Reinforcement Learning and Feature Extraction

Hang Zhao
College of Culture and Communication, Liming Vocational University, Quanzhou, 362000, China
E-mail: tarotworld@126.com

Keywords: animation production, mouth matching, reinforcement learning, feature extraction, actor-critic method

Received: May 14, 2024

With the development of the times, animation production has become increasingly sophisticated. Mouth matching is one of the key points to ensure the vividness and realism of animated characters. Therefore, this study proposed an animation character mouth matching model, which is based on Actor-Critic method of reinforcement learning. This model was combined with audio feature extraction and facial action coding to predict and generate animation character mouth matching synchronized with speech. This model was optimized by strategy gradient algorithm, aiming to achieve realistic and emotion-rich animated mouth matching. RAVDESS, VIDTIMIT, and SAVEE were selected as experimental datasets. Accuracy, F1 value, and peak signal-to-noise ratio were selected as performance indicators. GRU, CNN+GRU, and CNN+LSTM were selected as experimental comparison algorithms. The experimental results showed that the proposed model had an average accuracy of 95.61% and an F1 value of 97.13% on three databases. Meanwhile, the peak signal-to-noise ratio and structural similarity index of the proposed model were 41.77 and 0.93, respectively, which were better than the comparative methods. In addition, the study tested the error of mouth shape under different emotions, and the results showed an average mean square error of only 6.639. Finally, the user survey results showed that the animated characters generated by the proposed model received more recognition in mouth shape matching and realism, with a highest selection rate of 98.64%. The successful application of the proposed model provides new ideas and methods for research in related fields, laying the foundation for further promotion and innovation of animation production technology

Povzetek: Raziskava predstavlja model za usklajevanje ust animiranih likov, ki temelji na metodi Actor-Critic v okviru stojnega učenja. Model uporablja ekstrakcijo značilnosti zvoka in obraznih potez za sinhronizirano in realistično gibanje ust, kar omogoča izražanje čustev.

1 Introduction

In today's digital age, animation production plays an increasingly important role in the entertainment industry. Animation production not only provides viewers with rich and colorful visual enjoyment, but also becomes an important medium for people to express creativity and emotions [1, 2]. With the continuous advancement of technology, animation production is no longer limited to simple visual presentation, but requires deeper shaping of animated characters, especially in terms of expression and communication [3, 4]. In addition, the charm of animation is far beyond just the visuals. Dubbing and music, as the emotional guide, deepen the emotional resonance of animation works [5, 6]. In this context, matching the mouth shape of animated characters has become a challenging task, which is directly related to the expression and emotional communication of animated characters [7, 8]. However, it is extremely complex to generate synchronized mouth movements, involving multiple challenges such as video generation, precise matching from audio to video generation, and training the model's generalization ability. Traditional mouth shape matching methods should meet the needs of dynamic scenes and complex emotional expression [9, 10]. Generating synchronized mouth movements with emotional features is always a challenge in computer animation research. Therefore, this study proposes an innovative animation character mouth matching model that takes into account reinforcement learning and feature extraction. This study focuses on the mouth design model of 3D character animation and speech emotion synchronization algorithm. The Actor-Critic method in reinforcement learning is introduced to continuously optimize and improve the performance of the model in learning mouth generation. Meanwhile, the feature extraction of audio and facial Action Units (AUs) ensures that the generated mouth shape is not only realistic, but also able to express rich emotional features. The goal of the research is not only to achieve the production of animated characters with real emotions and mouth shape matching, but also to promote the development of the entire animation production technology. This innovative research not only provides audiences with a richer audio-visual experience, but also promotes the digital entertainment industry. This leads animation production into a more fascinating era.

The study is divided into five parts. The first part introduces the current research on different methods of animation production and mouth matching worldwide. The different applications of the Actor-Critic method are introduced too. The second part mainly introduces animation production, animated characters, and other content. The third part provides an in-depth introduction to the methods of mouth shape matching models for animated characters. The fourth part conducts experiment on the performance of the mouth shape matching model proposed in the study to verify its feasibility. The last part is a summary and discussion of the article.

2 Related works

The advancement of technology has provided strong support for the development of the animation industry. With the continuous advancement of computer and network technology, the cost and cycle of animation production have gradually decreased, while also improving the efficiency and quality of animation production [11]. There are also many studies on animation production around the world. Ye et al. proposed a new method based on support vector machine and Augmented Reality (AR) animator system to address the challenge of creating virtual AR animated characters that closely interacted with the real environment. This allowed users to easily create in situ character animations that closely interacted with different real-world environments [12]. Arshad et al. proposed a method to explore the basic process of character assembly systems in 3D animation production, addressing the flexibility limitations in the character assembly process. This study suggested that chain assembly systems were the best choice for the animated characters. Therefore, the assembly process in the animation directly affected the actions and poses of the character in the final animation [13]. Paier et al. proposed a new hybrid animation framework to tackle the challenging task of creating realistic facial animations in computer graphics. This framework utilized the latest advances in deep learning to provide an interactive animation engine for facial expression editing through simple and intuitive visualization [14]. Facial areas, such as eyes or teeth, could not be stably synthesized in the generation of animated characters. Then, K. Gu et al. proposed a landmark-driven dual-flow network to obtain data from multi-source images to achieve facial region synthesis and learn more details, thus effectively improving the fidelity of animated characters [15].

The advantage of the Actor-Critic method is that it has high efficiency and scalability when dealing with large state spaces [16]. In practice, the Actor-Critic method has been widely applied in fields such as robot control and game intelligence, which has achieved significant results. Hong et al. proposed a dual time scale stochastic approximation algorithm for a two-layer optimization problem, which included an external objective function and an internal strongly convex problem. The convergence speed of the Actor-Critic method was fast, providing an effective method for solving global optimal strategies and double-layer optimization problems [17]. Xi et al. proposed a new control strategy to address the shortcomings of traditional automatic power generation control in dealing with strong stochastic disturbances caused by renewable energy infiltration. The Actor-Critic method and incentive heuristic mechanism were introduced to improve the system control performance, improve the dynamic performance of the power system, and achieve regional optimal coordinated control [18]. Han et al. proposed a control method based on the Actor-Critic method for a reinforcement learning framework to address the stability of model free reinforcement learning in robot control tasks. Through empirical evaluation, the learned strategies could to some extent restore the system to a balanced state or path point when the system was disturbed by uncertain factors. This demonstrated its advantages in improving system robustness and stability [19]. Zhong et al. proposed a dynamic multi-channel access framework based on deep Actor-Critic reinforcement learning to effectively utilize limited spectrum resources. This considered the situation where both single and multiple users attempted to access the channel simultaneously, effectively improving the resource utilization [20].

The summary of related works related to the research content is shown in Table 1.

Table 1: Summary of related works

Method	Result	Disadvantage	Reference
Animation character creation based on support vector machine and AR animated system	Users can easily create in-situ character animations that interact closely with different real-world environments	Not paying attention to animation character details	Ye et al.
Using chain assembly system to personify 3D animated characters	The animated characters created are more realistic and the creation efficiency is improved.	Not paying attention to animation character details	Arshad et al.

Building a facial expression editing method using deep learning	It is easy and intuitive visualization for facial expression editing.	The neural texture model used does not consider the view dependence of texture.	Paier et al.
Landmarks-driven two-stream networks that obtain data from multiple source images.	The fidelity of animated characters is improved.	Large computing resources and long time consuming.	Gu et al.
A dual time scale stochastic approximation algorithm framework	The Actor-Critic method has faster convergence rate, which solves the dual optimization and provides an efficient method	The Actor-Critic method is not applied to a specific domain.	Hong et al.
Introduce Actor-Critic method and incentive mechanism	The system control performance is improved, the dynamic performance of the power system is improved, and the regional optimal coordinated control is realized.	High computational complexity.	Xi et al.
Control method of reinforcement learning framework based on Actor-Critic method	When the system is disturbed by uncertain factors, it can restore the system to the equilibrium state or path point to a certain extent	The method is very sensitive to the choice of hyperparameters.	Han et al.
Dynamic multi-channel access framework based on deep Actor-Critic reinforcement learning	The resource utilization is improved.	The stability of the algorithm is poor.	Zhong et al.

In summary, the development of animation production is becoming increasingly technological, and the Actor-Critic method is also applied in many fields. However, there is limited research on mouth shape generation. Some methods generate animated videos by selecting frames from specific character databases and combining them. However, these methods highly rely on specific roles and incur significant costs when transitioning to new speakers. Therefore, this study proposes an animation character mouth matching model based on the Actor-Critic method. Then, the generation of synchronized mouth movements is regarded as a sequence generation, achieving the goal of character independence by matching speech with corresponding features. The innovation of the research includes the generalization of the model, which can be generalized across different languages and speaker roles. Next is to expand to 2D to 3D, introducing 2D video datasets to compensate for the shortcomings of 3D character datasets. Finally, emotional feature extraction is performed by analyzing audio emotional features to recreate facial micro changes. This is to reflect the contextual information and emotional intensity in the input speech.

3 The development history of animation production and animated characters

Animation is an art and technical form that creates motion effects by continuously playing a series of still images or graphics [21]. Animation has become an indispensable part of the global cultural industry, providing audiences with endless creativity and entertainment [22]. Animation production is a complex process involving multiple steps, from concept and storyboard design, to character modeling and scene layout, and then to the creation of keyframe animation and interval frames [23]. This process also includes audio design, animation rendering, and post-processing. The final animated work can be published and distributed on media platforms such as movies, television, and the internet. The origin of animation can be traced back to the late 19th century, and its development history is summarized as shown in Figure 1.

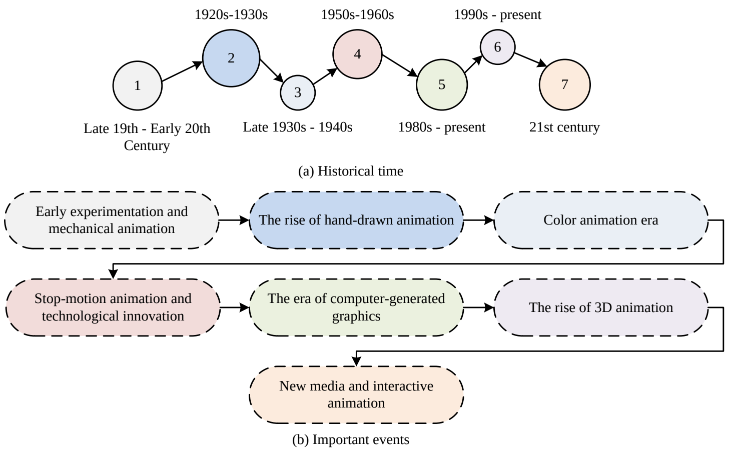

1920s-1930s
1950s-1960s
1990s - present
1
2
3
4
5
6
7
Late 19th - Early 20th Century
Late 1930s - 1940s
1980s - present
21st century
(a) Historical time
Early experimentation and mechanical animation
The rise of hand-drawn animation
Color animation era
Stop-motion animation and technological innovation
The era of computer-generated graphics
The rise of 3D animation
New media and interactive animation
(b) Important events

Figure 1: Development history of animation production

In Figure 1 (a), animation production has roughly gone through 7 historical stages and key developments. Figure 1 (b) shows seven key events in the development of animation production. Specifically, experimenters such as Émile Reynaud from France and J. from the United Kingdom Stuart Blackton attempted to conduct animation experiments using mechanical means such as rotating discs and slides during the early experimental and mechanical animation period (late 19th to early 20th century). The rise of hand drawn animation occurred from the 1920s to the 1930s. In 1928, Disney's Mickey Mouse became the first animation to synchronize sound and image, driving the mainstream development of hand drawn animation. The era of color animation emerged from the late 1930s to the 1940s, with Snow White in 1937 becoming the first full-length color animated film. The period of stop motion animation and technological innovation occurred from the 1950s to the 1960s, with 1955's Candy House being the first film to use stop motion animation technology. During this period, special effects animation films such as Disney's Cinderella (1950) and Mary Popins (1964) achieved great success.

The 1980s ushered in the era of computer-generated graphics. In 1982, Pixar's short film “Andre's Dream” became the first animation to use computer-generated graphics. Since the 1990s, the flourishing of 3D animation has become mainstream. Toy Story in 1995 was the first fully computer-generated full-length animated film, marking the dominant position of computer technology in animation production. During this period, many classic works emerged, including “The Lion King” (1994), “Frozen” (2013), and so on. In the 21st century, new media and interactive animation have emerged. With the popularization of the Internet and mobile devices, animation has entered a multi-platform era. New media forms such as interactive animation and virtual reality animation have gradually emerged. This evolution has witnessed the rapid progress of animation production technology and the innovative spirit of creators at different times. It is worth noting that animated characters play a crucial role in animation production, as shown in Figure 2.

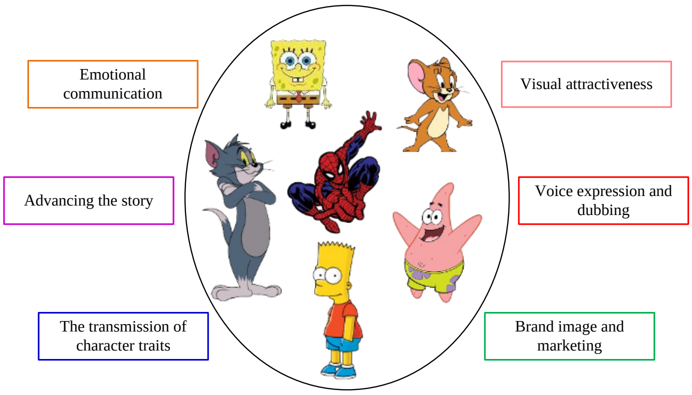

Emotional communication
Visual attractiveness
Advancing the story
Voice expression and dubbing
The transmission of character traits
Brand image and marketing

Figure 2: The key role of animated characters in animation

In Figure 2, animated characters are not only the main body of the story, but also key elements for emotional communication and visual attraction. Firstly, animated characters convey rich emotions through exquisite expressions and actions, allowing the audience to be more deeply immersed in the story plot. Secondly, as the driving force of the story, the conflicts, goals, and developmental trajectories of animated characters drive the entire story forward, increasing the depth and appeal of the story [26]. Each animated character is unique, showcasing various values and cultural characteristics through their words and actions, enriching the connotation of the entire work. Visually, the design of animated characters directly determines the attractiveness of a work. Clever images can attract the audience's attention, making the animation more engaging. In terms of voice, dubbing can express the personality and emotions of characters, giving them a unique timbre [27, 28]. In addition, some successful animated characters have become powerful brand images, surpassing the animation itself and driving the sales and successful marketing of related products [29]. Therefore, animated characters are not only characters, but also bridge the emotional connection between the audience and the work, contributing irreplaceable elements to the success and charm of animated works.

4 Establishment of mouth shape matching model for animated characters in action production

Animated characters play a crucial role in storytelling and visual appeal as key elements. Therefore, a mouth shape matching model is proposed for animated characters that makes them more realistic and vivid. The study aims to establish models from three aspects, namely audio feature extraction, facial feature extraction of animated characters, and the application of Actor-Critic methods.

4.1 Audio feature extraction and processing in animation character mouth matching model

When performing mouth shape matching in animation production, extracting key acoustic features from the input audio signal can enable the mouth shape matching model of animated characters. As a result, information, such as tone, speed, and emotion of speech, can be understood and captured. Therefore, the first step in constructing a model is to analyze the audio data. Then, the model can better adapt to the speech styles of different speakers and provide a foundation for subsequent mouth shape matching. The study selects Mel Frequency Cepstral Coefficients (MFCC) as the key method for audio feature extraction. The MFCC design is inspired by the way the human ear perceives sound, which can better simulate the perceptual characteristics of the human ear, making it more suitable for processing human speech signals. The MFCC extraction in the proposed model is shown in Figure 3.

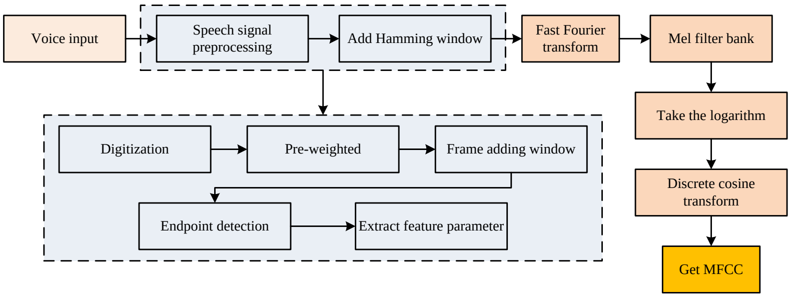


Figure 3: Flow chart of extracting MFCC

In Figure 3, to extract MFCC, the first step is to obtain a speech signal and input it. The preprocessing and Hamming window operation of the input speech audio signal is studied to distinguish between high-frequency signals and low-frequency signals. Then, the signal is converted to the frequency domain through fast Fourier transform. A Mel filter bank is used to uniformly distribute it on the Mel frequency scale, simulating human ear perception. The logarithm is taken to simulate nonlinear volume perception. Subsequently, the spectral signal is transformed into cepstral coefficients through discrete cosine transform, and the key MFCC is ultimately extracted. One advantage of MFCC is that it has relatively less interference with noise. Meanwhile, it is easier to display the non-uniform structure of frequency in feature representation through the nonlinear characteristics of Mel scale. There is a mapping relationship between the Mel frequency and the actual frequency of the signal. This is approximately linear in the frequency range below 1kHz and logarithmic in the high frequency range. The research establishes a relationship model between the two as shown in equation (1).

$$F_{mel} = 2595\log_{10}\left(1+\frac{F_t}{700}\right) \qquad (1)$$

In equation (1), F_{mel} and F_t represent the Mel frequency and actual frequency, respectively. In the preprocessing process, the audio signal needs to be processed in frames. The simplification effect of longer frames will be compromised due to the difficulty in extracting effective features from shorter frames.

Therefore, the entire audio is divided into a set of 20-40 millisecond frame segments. A pre-emphasis operation is adopted to counteract the suppression effect of the system on the high-frequency part of the speech signal. The processed effect is shown in Figure 4.

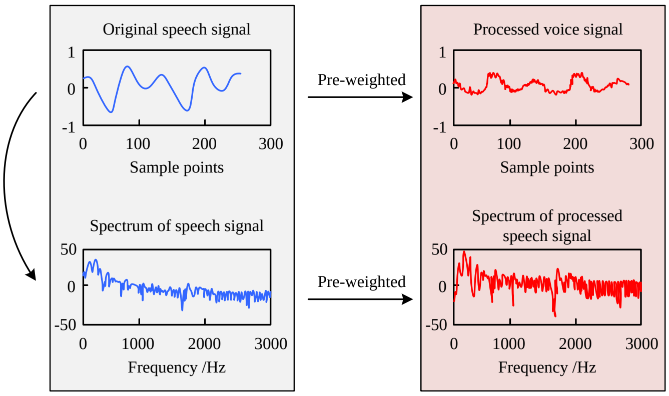


Figure 4: Contrast of signal and spectrum before and after pre-weighted processing

In Figure 4, the purpose of the pre-emphasis operation is to emphasize the high-frequency part and compensate for the high-frequency components in the speech signal that are suppressed by the system due to vocal fold and lip effects. A first-order limited length impulse response feedback filter is introduced to weight the speech signal, making the amplitude of the high-frequency part relatively enhanced. This can better capture high-frequency features in subsequent processing, improve the clarity and distinguishability of speech signals. Speech signals exhibit non-translational characteristics under the influence of changes in motion state. For the convenience of processing, this study divides longer speech signals into shorter speech signal segments and introduces frame shift to ensure a smooth transition in information processing. Frame shifting refers to a certain overlap between adjacent speech frames, rather than tightly connecting the beginning and end of the frames. This can ensure a smooth transition between two frames during the framing stage, avoiding the loss of key information. The combination of finite terms may cause Gibbs effect when using Fourier transform to analyze the spectral characteristics of frame signals. The Gibbs effect may deviate from the original speech signal after signal processing at the beginning and end of the frame. Therefore, a Hamming window is added. The Hamming window function is shown in equation (2).

$$\omega(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad (2)$$

In equation (2), $\omega(n)$ is the value of the window function at the discrete sequence n, and N is the length of the window. Window operation can minimize the amplification deviation caused by discontinuous parts in the application of fast Fourier transform, obtain a more continuous wave output, and reduce the influence of the Gibbs effect. This helps improve the accuracy of spectrum analysis. In addition, the characteristics of speech signals change over time in actual speech. Therefore, for each time t, a delta coefficient is calculated by weighting and summing the MFCC features of a certain number of frames before and after to reflect the dynamic characteristics of that time. The delta coefficient is shown in equation (3).

$$delta(t) = \frac{\sum_{i=1}^{I} n \cdot \left[MFCC(t+i) - MFCC(t-i)\right]}{2 \cdot \sum_{i=1}^{I} i^2} \quad (3)$$

In equation (3), $delta(t)$ represents the dynamic feature, i represents the cycle in the calculation, and $MFCC(\ )$ represents the MFCC feature vector at a certain time. I represents the size of the calculation window, which specifies how many frames of dynamic information to consider when calculating the delta coefficient. The processing of research helps to capture the temporal changes in speech signals, providing more comprehensive information for audio analysis of mouth movements.

4.2 Facial feature extraction and expression processing of animated characters

In addition to considering the matching of audio and mouth shape, this study focuses on the emotional features conveyed in speech to make the facial features of animated characters more realistic. Then, it reflects these emotional characteristics in the facial animation performance of the character. Therefore, the study introduces the Facial Action Coding System (FACS). FACS systematically encodes facial expressions by dividing different facial movements into basic elements called AU. The study first names and distinguishes various regions of the face, as shown in Figure 5.

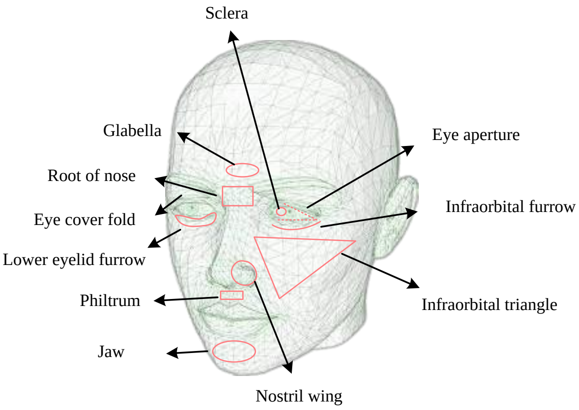


Figure 5: The naming and location of each area of the human face

Figure 5 shows that the human face is divided into 11 regions, namely sclera, glabella, root of nose, eye cover fold, lower eyelid furrow, jaw, nostril wing, infraorbital triangle, infraorbital furrow, and eye aperture. FACS contains multiple AUs, each representing the movement of muscles or muscle groups in different parts of the face. The combination of these AUs can form various facial expressions, making FACS a detailed and comprehensive system for recording and analyzing small changes in facial movements. The study lists some major AU codes and their meanings, as shown in Table 2.

Table 2: Main action units in FACS

Action unit	Description	Action unit	Description	Action unit	Description
AU1	Inner eyebrow raised	AU13	Pull the corners of the mouth up	AU25	Parted lips reveal teeth
AU2	Outer eyebrow raised	AU14	The corners of the mouth shrink towards the teeth	AU26	Parted lips see tongue
AU4	The eyebrows are generally lowered	AU15	Pull the corners of the mouth straight down	AU27	Parted lips see throat
AU5	Raise the eyelid	AU16	Pull the lower lip down	AU28	Suck the lips over the teeth
AU6	Lift the cheek	AU17	Squeeze the lower lip up to the top	AU41	Slightly lowered the lid
AU7	Eye contraction	AU18	Tuck the mouth in the middle	AU42	Lower the eyelids
AU9	Retraction and lifting nose	AU20	Pull the lips back	AU43	Close the eyes
AU10	Lift the upper lip	AU22	Curl the lips into a funnel	AU44	Lower eyelid up
AU11	Deepen the middle nasal lip	AU23	Clench the lips into one word	AU45	Blink the eyes
AU12	Turn up the corners of the mouth	AU24	Squeeze the lips together	AU46	Monocular blink

In Table 2, FACS encodes many subtle facial expressions, which is crucial for accurately interpreting and simulating subtle differences in facial expressions. Analyzing each expression yields a combination of AU. For example, the combination of happy expressions is AU6+AU12, which means lifting the cheeks and lifting the corners of the mouth. The combination of sad expressions is AU1+AU4+AU15, which means lifting the internal eyebrows, lowering the eyebrows as a whole, and pulling the corners of the mouth vertically downwards. The combination of surprise expressions is AU1+AU2+AU5+AU26, which means raising the inner eyebrows, raising the outer eyebrows, lifting the upper eyelids, and separating the lips to see the tongue. The combination of fear expressions is quite complex, consisting of AU1+AU2+AU4+AU5+AU7+AU20+AU26. It includes a combination of lifting the internal eyebrows, raising the external eyebrows, lowering the eyebrows as a whole, lifting the upper eyelids, contracting the eyes, pulling the lips backwards, and separating the lips to see the facial features of the tongue. Principal Component Analysis (PCA) is used to generate face grid after obtaining facial features when expressions change through FACS. Then, key features can be extracted from face data and a low-dimensional representation of faces can be generated. The goal of PCA is to find a transformation that projects the original data into a low-dimensional space while preserving the variance of the original data as much as possible. Firstly, facial feature points are preprocessed, as shown in equation (4).

$$x_c = x - \mu \tag{4}$$

In equation (4), x_c represents the data after centralization, x represents the feature vector of the original dataset, and μ represents the mean vector. The covariance matrix C for the centralized data is calculated, as shown in equation (5).

$$C = \frac{1}{n}\sum\nolimits_{i=1}^{n} x_{c,i}x_{c,i}^T \tag{5}$$

In equation (5), n represents the number of samples, and $x_{c,i}$ represents the i-th centralized sample. The eigenvalues of covariance matrix are decomposed to obtain the eigenvalues and corresponding eigenvectors. The eigenvector corresponding to the largest eigenvalue defining the main direction of the PCA space can be selected according to the size of the eigenvalues. The transformation matrix W is constructed using the selected eigenvectors. The original data are projected into the PCA space, as shown in equation (6).

$$x_{PCA} = W^T x_c \tag{6}$$

It is worth noting that the study constructs several assumptions when applying PCA. The assumptions include that the data changes are primarily linear, that the centralized feature points are independent of each other, that the eigenvalue distribution allows most of the variance to be preserved by selecting a small number of principal components, that the data need to be normalized to eliminate scaling effects, and that a certain reconstruction error is accepted. Therefore, the face Q is approximately represented as shown in equation (7).

$$Q = \overline{E} + \sum_{i=1}^{l} \alpha_i E_i \tag{7}$$

In equation (7), $\overline{E}$ represents the average face, E_i represents the i-th PCA face vector, and α_i is the coefficient. The study first needs to obtain the closest grid of neutral expressions in PCA space. The definition of feature point matching energy is shown in equation (8).

$$E_1 = \sum_{j=1}^{m_l} \left\| v_{i_j} - c_j \right\|^2 + \sum_{k=1}^{m_c} \left\| M v_{c_k} - s_k \right\|^2 \tag{8}$$

In equation (8), c_j represents the 3D position of feature point j, and v_{i_j} represents its mesh vertex. s_k represents the two-dimensional spatial feature points of the face, v_{c_k} represents the corresponding three-dimensional spatial feature points of the face, and M represents the projection matrix from two-dimensional to three-dimensional. The index of contour feature points on the grid can be determined through equation (8). The energy term for matching depth maps is defined as equation (9).

$$E_2 = \sum_{j=1}^{n_d} \left\| v_{d_j} - p_j \right\|^2 \tag{9}$$

In equation (9), v_{d_j} represents the vertex of the mesh, and p_j is the point closest to v_{d_j} in the depth map. The study aims to minimize energy through a series of least squares optimizations, which are iterated 5 to 8 times. Between consecutive iterations, update the mesh vertices corresponding to the contour feature points in equation (8) and the nearest point of each mesh vertex in equation (9). After obtaining the grid of neutral expressions, the grid of other expressions can be calculated based on the AU changes in FACD. Meanwhile, the grid deformation can use all facial feature points on color images as additional positional constraints.

4.3 The actor-critic method in animation character mouth matching model

The study regards the task of generating mouth animation as generating a sequence. Each time step corresponds to a specific facial expression or mouth state. The model uses the speech audio features and facial features of the character from the previous text as input states for the current time step, thereby predicting the facial features for the next time step.

This study uses the Actor-Critic method, which allows the model to gradually unfold the sequence of mouth animation generation. At each time step, the Actor generates a prediction of facial features, and the Critic evaluates the quality of this prediction and provides feedback. The schematic diagram of the Actor-Critic method is shown in Figure 6.

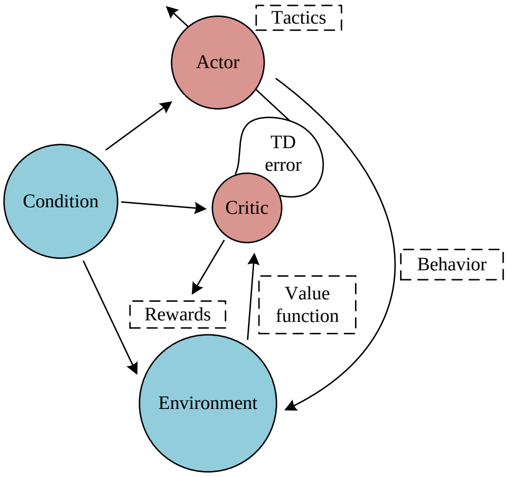


Figure 6: Actor-Critic process diagram

In Figure 6, firstly, the method begins training by initializing the parameters of the Actor network and the Critic network. Then, based on the current state, the Actor network is used to generate actions and execute them, observing rewards and new states. Next, a Critic network is used to calculate the value function or state action value function of the current state. It updates the parameters of the Actor network based on the policy gradient algorithm to maximize the expected reward. Meanwhile, the algorithm updates the parameters of the Critic network based on the value function to reduce estimation errors. This process will continue to repeat until convergence or termination conditions are met. Through this approach, the Actor-Critic method can gradually learn the optimal strategy and achieve maximum expected return. The strategy gradient formula is shown in equation (10).

$$\nabla_\theta J = \sum\nolimits_{s,a} \pi(a \mid s) \nabla_\theta \log \pi(a \mid s) Q(s,a) \quad (10)$$

In equation (10), J is a function of expected return, θ is a policy parameter, $\pi(a \mid s)$ is a policy, and $Q(s,a)$ is a state action value function. The policy gradient theorem is the basis of this formula, which states that the policy gradient is equal to the gradient of the product of the state action value function $Q(s,a)$ and the policy $\pi(a \mid s)$ with respect to the policy parameter θ. The gradient of expected returns with respect to policy parameters can be obtained by summing up the state action space. The formula for updating the value function is shown in equation (11).

$$\Delta V(s) = \sum\nolimits_a \pi(a \mid s)\left[Q(s,a) - V(s)\right] \quad (11)$$

In equation (11), $V(s)$ represents the state value function. This formula is derived based on the Bellman equation. The Bellman equation states that the state value function is equal to the weighted sum of the expected returns obtained by taking all possible actions in that state. The update amount $\Delta V(s)$ of the state value function can be obtained by calculating the difference between the state action value function $Q(s,a)$ and the state value function $V(s)$ and performing a weighted sum. The strategy update formula is shown in equation (12).

$$\theta \leftarrow \theta + \beta \nabla_\theta \log \pi(a \mid s) Q(s,a) \quad (12)$$

In equation (12), β represents the learning rate. This formula is derived based on policy gradients. The update direction of policy parameter θ can be obtained by calculating the policy gradient $\nabla_\theta \log \pi(a \mid s) Q(s,a)$ through research. Then, the learning rate β is used to control the update step size and updates the policy parameters along the gradient direction. In this way, the study can gradually adjust strategies to maximize expected returns. In summary, the overall process of the animation character mouth shape matching model based on the Actor-Critic method is shown in Figure 7.

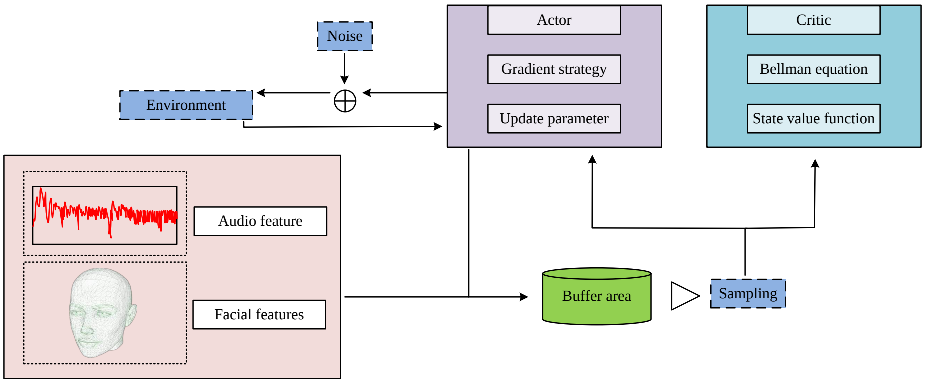


Figure 7: Animation character mouth matching model

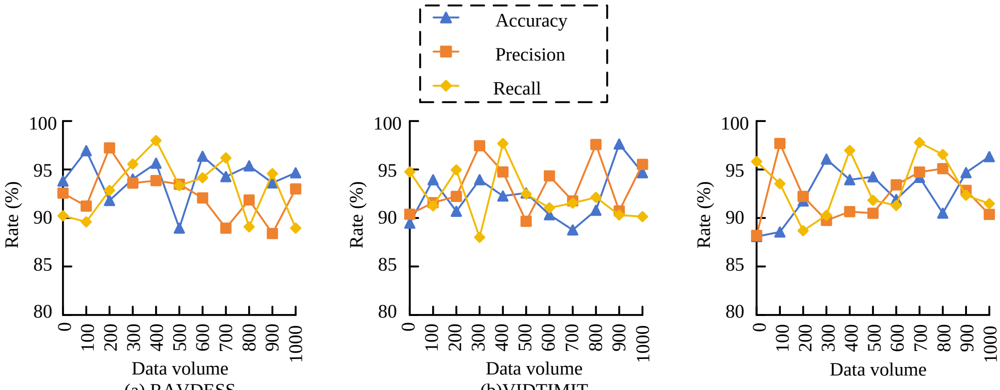
In Figure 7, a parameterized Actor model is trained to generate a sequence. The state includes audio features and previously generated expression features, while the action is the expression feature for the next time step. In addition, a parameterized Critic model was introduced to guide the Actor model in generating more realistic mouth movements through rewards. During the training process, real sequence data and synthetic sequence data are used to optimize the Critic model. To improve performance, a replay buffer is introduced and a deep deterministic policy gradient algorithm is used for training to minimize the error of the Critical network.

5 Performance verification and actual effect analysis of animation character mouth matching model

To explore the effectiveness and superiority of the research content, this study conducted experimental verification on the mouth shape matching model of animated characters. Firstly, it compared the performance of the model on different datasets, and then compared the different methods with the generated renderings of the model.

5.1 Performance verification of mouth shape matching model for animated characters

To verify the effectiveness and superiority of the mouth shape matching model for animated characters, this study selected three databases: Ryerson Audio Visual Database of Emotional Speech and Song (RAVDESS), Video Limit (VIDTIMIT), and Surrey Audio Visual Express Emotion (SAVEE) from the University of Surrey in the UK. RAVDESS is a database used for studying emotion recognition, containing audio and video data from different actors. The voice and song samples in this database cover different emotional states, such as joy, sadness, anger, etc. VIDTIMIT is a database used for studying audio and video synthesis. It is an extension of the TIMIT database, providing video data on facial and lip movements corresponding to speech. SAVEE contains speech samples from different emotional states in English, used for emotion recognition and related research. The study randomly selected 3/5 of the data in the database as the training set, and the remaining 2/5 of the data as the testing set. The training results are shown in Figure 8.



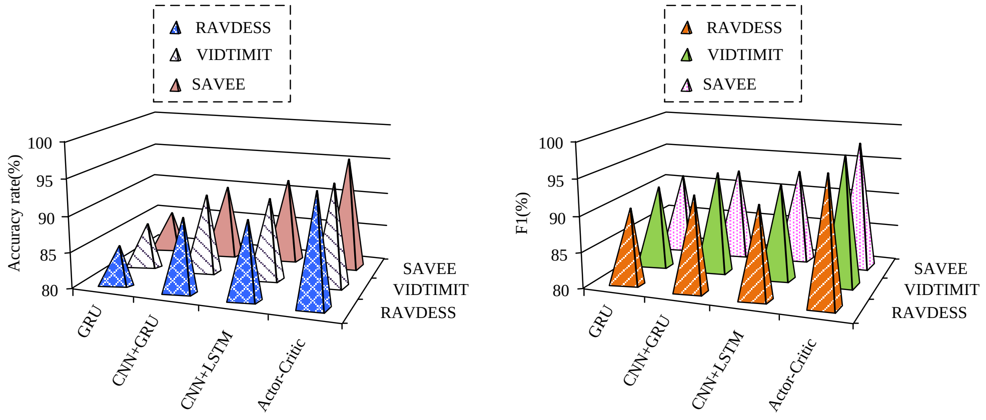
(a) RAVDESS

(b)VIDTIMIT

(c)SAVEE

Figure 8: Comparison of training results

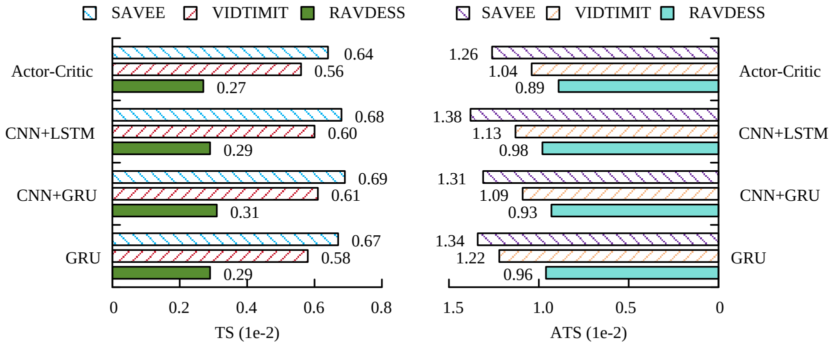
Figure 8 (a) shows that the training accuracy, precision, and recall of the RAVDESS database are all above 90%. Figure 8 (b) shows that the training accuracy and recall of the VIDTIMIT database are both above 90%. The accuracy is 89.56% when the data volume is 900, and the rest are above 90%. Figure 8 (c) shows that the training results of the SAVEE database are also excellent, with accuracy, precision, and recall basically above 90%. Therefore, this study further examined the performance indicators on the test set and compared them with mouth shape matching models supported by Gated Recurrent Unit (GRU), Convolutional Neural Network+Gated Recurrent Unit (CNN+GRU), and CNN+Long Short-Term Memory (CNN+LSTM) models. GRU is suitable for processing sequential data and is able to capture long-term dependencies in time series. CNN+GRU combines convolutional neural networks (for extracting visual features) and GRU (for processing time series features) to improve the accuracy of mouth matching. CNN+LSTM replaces the GRU with a Long Short-Term Memory (LSTM), which is equally good at processing sequence data and is able to learn long-term dependencies on information. In the experiment, the key hyperparameter settings are as follows. The evaluation frequency is to evaluate the model performance every 1000 training steps. The learning rate of the Adam optimization algorithm is adjusted every 3×10^4 steps. The batch size is 256. The discount factor is used to determine the current value of future rewards, which is set to 0.99. The temperature parameter is used to control the updating amplitude of the policy gradient, which is set to 0.005. The comparison between accuracy and F1 value is shown in Figure 9.



(a) Accuracy comparison (b) F1 value comparison

Figure 9: Comparison results of accuracy and F1 value of four algorithms

Figure 9 (a) shows that for accuracy, GRU is below 85%. CNN+GRU and CNN+LSTM are basically around 90%. Among them, CNN+GRU has the highest accuracy on the VIDTIMIT database, reaching 90.67%, and CNN+LSTM has the highest accuracy on the SAVEE database, reaching 91.34%. The accuracy of the Actor-Critic method proposed in the study is superior to the first three, reaching a maximum of 96.48%, with an average of 95.61% on the three databases. In Figure 9 (b), the F1 value of the Actor-Critic method is also higher than that of the three comparison algorithms, with an average of 97.13%. Furthermore, the study compared the time series analysis performance of the four algorithms using Temporal Smoothness (TS) and Adaptive Temporal Smoothness (ATS) as indicators. The results are shown in Figure 10.



(a) TS (b) ATS

Figure 10: Comparison of TS and ATS

In Figure 10 (a), the Actor-Critic method has the smallest TS in all three databases, with values of 0.64, 0.56, and 0.27, respectively. In Figure 10 (b), the ATS of the Actor-Critic method is also the smallest, with values of 1.26, 1.04, and 0.89, respectively. This means that the Actor-Critic method has a smoother temporal variation, which helps to reduce noise and abrupt changes, making the mouth matching results more stable and coherent. The study continued to compare the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) of the four algorithms, as shown in Figure 11.

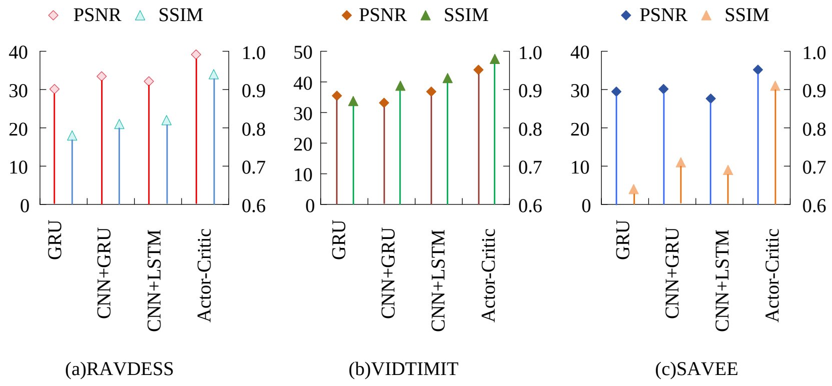


Figure 11: Comparison of PSNR and SSIM

Figure 11 (a) shows that on the RAVDESS database, the Actor-Critic method has a PSNR of 39.67 and a SSIM of 0.93. Figure 11 (b) shows that on the VIDTIMIT database, the Actor-Critic method has a PSNR of 46.58 and a SSIM of 0.96. According to Figure 11 (c), on the SAVEE database, the PSNR of the Actor-Critic method is 37.06 and the SSIM is 0.91. The results indicate that the Actor-Critic method has better image quality and more accurate preservation of structure and details compared to the comparative algorithms. Finally, the accuracy of the four methods is compared, and the results are shown in Table 3.

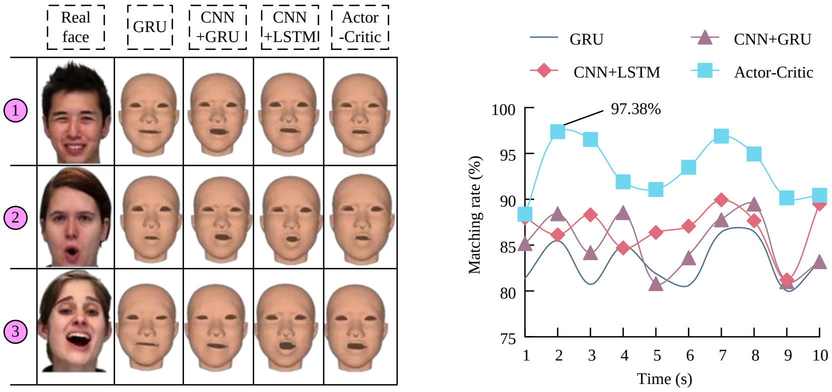
Table 3: Accuracy comparison of the four methods

-	Accuracy (%)			
	GRU	CNN+GRU	CNN+LSTM	Actor-Critic
RAVDESS	89.46	91.24	92.43	96.85
VIDTIMIT	88.52	91.94	93.02	95.97
SAVEE	87.46	90.61	92.99	97.05
Mean value	88.48	91.26	92.81	96.62

From Table 3, the Actor-Critic method has the highest accuracy on the three datasets, reaching 96.62% on average. The average accuracy of GRU, CNN+GRU, and CNN+LSTM is 88.48%, 91.26% and 92.81%, respectively. Therefore, the mouth matching model based on Actor-Critic method has high precision, F1 value, and accuracy rate, which has important practical application significance in the animation industry. It can generate highly synchronized and emotion-rich animation mouth shapes with speech, significantly improving the animation quality and enhancing the expressiveness of characters.

5.2 Analysis of Actual Effects of Animation Character Mouth Matching Model

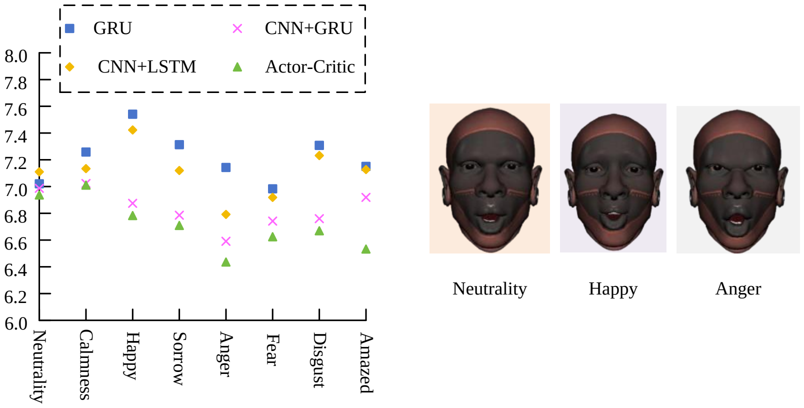
To further demonstrate the superiority of the animation character mouth matching model supported by the Actor-Critic method proposed in the study, the effect images generated by various comparison algorithms were compared. The matching effect and matching rate of the animation character's mouth shape matching effect are shown in Figure 12.



(a) Effect drawing (b) Matching rate

Figure 12: Animated characters' mouth matching effect

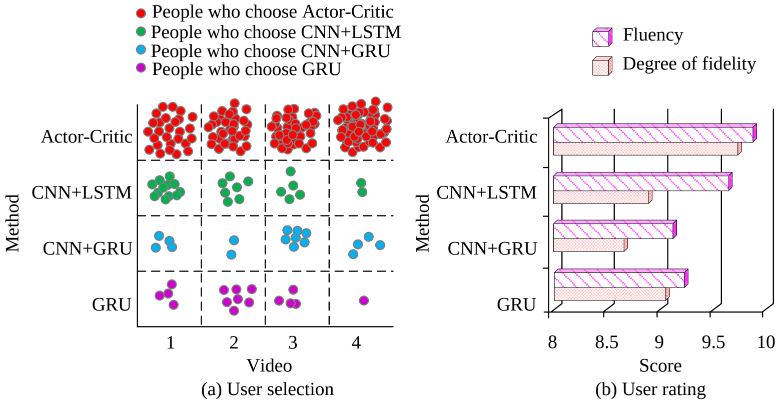
Figure 12 (a) shows that the animated character facial model generated by the Actor-Critic method is more realistic and more similar to the expressions of real faces. Figure 12 (b) calculates and compares the matching rate of the matching effect map within 10 seconds. Figure 12 (b) shows that the matching rate of the Actor-Critic method is consistently better than the four comparison algorithms. The highest matching rate is achieved in the second, reaching 97.38%, with an average matching rate of 94.29%. The average matching rate of GRU renderings is 82.54%. The average matching rates of CNN+GRU and CNN+LSTM renderings are 84.97% and 83.93%, respectively. To further understand the performance of each model on the test set, this study calculated the mean square error of parameters corresponding to different emotions and facial data. This study provided facial models that said the same sentence under three different emotions. This is to better observe the changes in facial expressions, as shown in Figure 13.



(a) Mean square error (b) Different emotional mouth movements

Figure 13: Animated characters' mouth matching effect

Figure 13 (a) shows that, overall, the mean square error of the Actor-Critic method is consistently lower than other comparison methods. When the emotion is anger, the mean square error is the smallest, at 6.436. This is because angry expressions contain more AUs and facial features are more prominent. The average mean square error of the Actor-Critic method is only 6.639. Figure 13 (b) shows the differences in facial expressions of the same sentence under different emotions. This indicates that the Actor-Critic method not only has a higher mouth shape matching rate, but also a more realistic expression. Finally, the study presented users with four animated videos generated by each of the four methods and asked them to choose videos that were more natural and realistic. Meanwhile, each video was rated. The results are shown in Figure 14.



(a) User selection (b) User rating

Figure 14: User selection and rating results

Figure 14 (a) shows that users believe that the mouth shape matching and facial expressions of animated characters in videos generated by the Actor-Critic method are more vivid and realistic. After careful statistics, more than 90% of users choose the Actor-Critic method, and the selection rate of video number 4 even reaches 98.64%. Figure 14 (b) shows that for the videos generated by the Actor-Critic method, the fluency and realism ratings reached 9.87 and 9.72, respectively. However, in the comparison algorithm, the highest flow smoothness is only 9.64. For videos from CNN+LSTN, the highest fidelity is only 9.04, while for videos from GRU. This indicates that the Actor-Critic method has achieved significant success in mouth shape matching and character expression generation, which has been highly recognized by users. This can demonstrate the effectiveness and superiority of the proposed animation character mouth shape matching model.

6 Discussion

The model of mouth matching was put forward. First of all, from the perspective of performance indicators, the proposed mouth matching model was superior to the methods listed in Table 1 in terms of accuracy, F1 value, PSNR, and SSIM. Specifically, the average accuracy of the research model on the three databases reached 95.61%, the F1 value was 97.13%, the PSNR was about 41.77, and the SSIM was about 0.93. In contrast, other methods in Table 1 had lower performance on these indicators, for example, the method using support vector machine and AR animator system [12], and the method using chain assembly system [13]. Although they improve the authenticity of animated characters, they do not pay special attention to the accuracy of mouth matching and the richness of emotional expression. The reasons for the difference in performance can be attributed to several key factors. The proposed model adopted Actor-Critic method and combined audio feature extraction and facial action coding, which not only improved the accuracy of mouth matching, but also enhanced the richness of emotional expression. In addition, 2D video datasets were introduced to make up for the shortcomings of 3D character datasets, helping to improve the generalization ability of the model. The advantages of the proposed model are that it can optimize mouth shape generation by Actor-Critic method, improve the learning and prediction ability of the model, and capture the emotional features in audio and reflect them in the facial animation of animated characters. However, the model mainly focuses on mouth matching, which does not deeply analyze other details such as blinking and turning the head. In summary, the proposed model provides a new solution for animation production by introducing reinforcement learning and feature extraction techniques. It not only improves the accuracy of mouth matching and the emotional expression ability of animated characters, but also lays a foundation for the further promotion and innovation of animation production technology.

In addition, MFCC was used as the key method for audio feature extraction. However, MFCC may be sensitive to noise, whose sensitivity to tone change may lead to feature distortion in speech processing with different intonation or pitch. At the same time, MFCC has limited ability to capture nonlinear features and has a certain dependence on audio quality. Therefore, MFCC can be replaced by Perceptual Linear Prediction (PLP), another feature extraction method simulating auditory characteristics of human ears. PLP takes the perceptual nonlinearity into account in the extraction. Therefore, PLP may have better ability to capture the nonlinear features of speech signals.

When using the Actor-Critic method, the research did not deeply consider the setting of hyperparameters in the Actor-Critic method. The parameter setting in the research referred to existing literature. Therefore, the subsequent research can further carry out a detailed analysis of the hyperparameter setting scheme. In addition, as the size of the dataset increases, the calculation requirements for the Actor-Critic method are also the same. The scalability of the model needs to be considered to handle larger datasets. For example, the model can be trained on multiple Gpus simultaneously through data parallelization techniques, or the model parallelization method can be used to distribute different parts of the model to different computing devices. In addition, incremental learning allows models to be updated gradually as new data arrive, rather than being trained from scratch each time, which not only increases efficiency but also reduces the waste of computational resources. Distributed training frameworks can also enable models to be trained on multiple compute nodes to handle larger datasets. The performance of the model can be maintained or improved through these methods. The computational complexity can be effectively managed and reduced to ensure that the model can adapt to the growing volume of data.

As automation advances, technologies must ensure that innovation is accompanied by appropriate ethical guidelines and regulatory measures to prevent misuse of technology, protect personal privacy, clarify liability, and promote the healthy development of technology. In addition, the use of automated methods to create realistic animation requires consideration of how to maintain the artistic and creative nature of animation. The automated methods should ensure that the spread of technology does not exacerbate social inequalities and update education and training systems to adapt to technological change. With these measures, future research can make better use of automated lip matching technology, while ensuring that it is applied ethically and responsibly.

7 Conclusion

A mouth shape matching model based on reinforcement learning and feature extraction is proposed to address the issue of mouth shape matching in animation production. The study first explores the development history of animation production and illustrates the importance of animated characters. Then, in this context, audio feature extraction, facial feature extraction, and Actor-Critic method are combined to complete the mouth shape matching of animated characters. Meanwhile, its final experimental verification is carried out. The experimental results showed that the model training results were good, with accuracy, precision, and recall basically above 90%. On the test set, the average accuracy and F1 value of the Actor-Critic method were better than GRU, CNN+GRU, and CNN+LSTM, reaching 95.61% and 97.13%, respectively. In addition, the Actor-Critic method had the smallest TS in all three databases, with values of 0.64, 0.56, and 0.27, respectively. ATS was also the smallest, with values of 1.26, 1.04, and 0.89, respectively. The average PSNR and SSIM of the Actor-Critic method on the three databases were approximately 41.77 and 0.93, respectively. Finally, a user survey was conducted, and over 90% of users chose the Actor-Critic method. Overall, this study provides an advanced mouth shape matching model for the animation production, making significant contributions to improving the expressiveness and realism of animated characters. Although the research delves into the emotional changes in mouth matching, the simulation of key details such as blinking and turning of the head has not been addressed. Future research can focus on simulating the natural rhythm of blinking and the fluency of head movements to enhance the realism of animated characters. In addition, there are important directions to improve the naturalness of character animation, including capturing facial micro-expressions, developing context-aware action generation algorithms, and realizing personalized and real-time action generation. At the same time, multi-modal emotion analysis, user interaction customization, and cross-cultural emotion expression research will further enrich the character's emotional level and expression.

Funding statement

The research is supported by: This work was supported (in part) by the Scientific research start-up fund project for introducing high-level talents from Liming Vocational University, under Grant (No. 24YG01S).

Conflict of interest

There is no conflict of interest between the authors.

References

[1] Y. Deng, "Fluid equation-based and data-driven simulation of special effects animation," Advances in Mathematical Physics, vol. 2021, no. 4, pp. 1-11, 2021. https://doi.org/10.1155/2021/7480422

[2] J. Zhang, G. Liao, and N. Li, "Combining active learning and local patch alignment for data-driven facial animation with fine-grained local detail," Neurocomputing, vol. 398, no. 7, pp. 431-441, 2020. https://doi.org/10.1016/j.neucom.2020

[3] W. Paier, A. Hilsmann, and P. Eisert, "Interactive facial animation with deep neural networks," IET Computer Vision, vol. 14, no. 6, pp. 359-369, 2020. https://doi.org/10.1049/iet-cvi.2019.0790

[4] S. Long, S. Andreopoulos, S. Patterson, J. Enkinson, and D. P. Ng, "Metabolism in motion: Engaging biochemistry students with animation," Journal of Chemical Education, vol. 98, no. 5, pp. 1795-1800, 2021. https://doi.org/10.1021/acs.jchemed.0c01498

[5] L. Wang, and J. Kim, "Exploring the caricature style identification and classification using convolutional neural network and human-machine interaction under artificial intelligence," International Journal of Humanoid Robotics, vol. 19, no. 3, pp. 26-38, 2022. https://doi.org/10.1142/S0219843622400096

[6] W. Paier, A. Hilsmann, and P. Eisert, "Example-based facial animation of virtual reality avatars using auto-regressive neural networks," IEEE Computer Graphics and Applications, vol. 41, no. 4, pp. 52-63, 2021. https://doi.org/10.1109/MCG.2021.3068035

[7] R. Ploetzner, S. Berney, and M. Bétrancourt, "A review of learning demands in instructional animations: The educational effectiveness of

animations unfolds if the features of change need to be learned," Journal of Computer Assisted Learning, vol. 36 , no. 6, pp. 838-860, 2020. https://doi.org/10.1111/jcal.12476

[8] C. A. Sanchez, and K. Weber, "Using relevant animations to counter stereotype threat when learning science," Journal of Applied Research in Memory and Cognition, vol. 8, no. 4, pp. 463-470, 2019. https://doi.org/10.1016/j.jarmac.2019.08.003

[9] X. Xu, "Multimedia VR image improvement and simulation analysis based on visual VR restructuring algorithm," Informatica, vol. 48, no. 1, 2024. https://doi.org/10.31449/inf.v48i1.5368

[10] A. Berg, D. Orraryd, A. J. Pettersson, and M. Hulténl, "Representational challenges in animated chemistry: self-generated animations as a means to encourage students' reflections on sub-micro processes in laboratory exercises," Chemistry Education Research and Practice, vol. 20, no. 6, pp. 710-737, 2019. https://doi.org/10.1039/C8RP00288F

[11] M. Hanif, "The development and effectiveness of motion graphic animation videos to improve primary school students' sciences learning outcomes," International Journal of Instruction, vol. 13, no. 3, pp. 247-266, 2020. https://doi.org/10.29333/iji.2020.13416a

[12] H. Ye, K. C. Kwan, W. Su, and H. Fu, "ARAnimator: In-situ character animation in mobile AR with user-defined motion gestures," ACM Transactions on Graphics, vol. 39, no. 4, pp. 1-83, 2020. https://doi.org/10.1145/3386569.3392404

[13] M. R. Arshad, K. H. Yoon, A. A. A. Manaf, and M. A. Ghazali, "Physical rigging procedures based on character type and design in 3D animation," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 3, pp. 4138-4147, 2019. https://doi.org/10.35940/ijrte.C5484.098319

[14] W. Paier, A. Hilsmann, and P. Eisert, "Interactive facial animation with deep neural networks," IET Computer Vision, vol. 14, no. 6, pp. 359-369, 2020. https://doi.org/10.1049/iet-cvi.2019.0790

[15] K. Gu, Y. Zhou, and T. Huang, "Flnet: landmark driven fetching and learning network for faithful talking facial animation synthesis," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, pp. 10861-10868, 2020. https://doi.org/10.1609/aaai.v34i07.6717

[16] J. Li, R. Wang, and K. Wang, "Service function chaining in industrial Internet of things with edge intelligence: A natural actor-critic approach." IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 491-502, 2022. https://doi.org/10.1109/TII.2022.3177415

[17] M. Hong, H. T. Wai, Z. Wang, and Z. Yang, "A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic," SIAM Journal on Optimization, vol. 33, no. 1, pp. 147-180, 2023. https://doi.org/10.1137/20M1387341

[18] L. Xi, J. Wu, Y. Xu, and H. Sun, "Automatic generation control based on multiple neural networks with actor-critic strategy," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 6, pp. 2483-2493, 2020. https://doi.org/10.1109/TNNLS.2020.3006080

[19] M. Han, L. Zhang, J. Wang, and W. Pan, "Actor-critic reinforcement learning for control with stability guarantee," IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 6217-6224, 2020. https://doi.org/10.1109/LRA.2020.3011351

[20] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 4, pp. 1125-1139, 2019. https://doi.org/10.1109/TCCN.2019.2952909.

[21] S. Starke, Y. Zhao, F. Zinno, and T. Komura, "Neural animation layering for synthesizing martial arts movements," ACM Transactions on Graphics, vol. 40, no. 4, pp. 1-16, 2021. https://doi.org/10.1145/3450626.3459881

[22] P. Preethi, and H. R. Mamatha, "Region-based convolutional neural network for segmenting text in epigraphical images," Artificial Intelligence and Applications, vol. 1, no. 2, pp. 119-127, 2023. https://doi.org/10.47852/bonviewAIA2202293

[23] H. Chen, D. Zhao, and J. Barbič, "Capturing animation-ready isotropic materials using systematic poking," ACM Transactions on Graphics, vol. 42, no. 6, pp. 1-27, 2023. https://doi.org/10.1145/3618406

[24] J. Wolper, Y. Fang, M. Li, J. Lu, M. Gao and C. Jiang, "CD-MPM: continuum damage material point methods for dynamic fracture animation," ACM Transactions on Graphics, vol. 38, no. 4, pp. 1-15, 2019. https://doi.org/10.1145/3306346.3322949

[25] H. Eom, D. Han, J. S. Shin, and J. Noh, "Model predictive control with a visuomotor system for physics-based character animation," ACM Transactions on Graphics, vol. 39, no. 1, pp. 1-11, 2019. https://doi.org/10.1145/3360905

[26] B. John, S. Jörg, S. Koppal, and E. Jain, "The security-utility trade-off for iris authentication and eye animation for social virtual avatars," IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 5, pp. 1880-1890, 2020. https://doi.org/10.1109/TVCG.2020.2973052

[27] M. Guo, F. Xu, S. Wang, Z. Wang, M. Lu, X. Cui, and X. Ling, "Synthesis, style editing, and animation of 3D cartoon face," Tsinghua Science and Technology, vol. 29, no. 2, pp. 506-516, 2024. https://doi.org/10.26599/TST.2023.9010028

[28] W. Carlson, R. Hackathorn, and R. Parent, "Computer graphics and animation at the ohio state university," IEEE Computer Graphics and

Applications, vol. 41, no. 3, pp. 8-17, 2021. https://doi.org/10.1109/MCG.2021.3070624

[29] M. Brehmer, B. Lee, P. Isenberg, and E. K. Choe, "A comparative evaluation of animation and small multiples for trend visualization on mobile phones," IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 1, pp. 364-374, 2020. https://doi.org/10.1109/TVCG.2019.2934397

[30] Z. Ma, C. Li, X. Liu, H. Wu, and Z. Wen, "Separating shading and reflectance from cartoon illustrations," IEEE Transactions on Visualization and Computer Graphics, vol. 30, no. 7, pp. 3664-3679, 2024. https://doi.org/10.1109/TVCG.2023.3239364

The Application of Multiple Regression Model in Blended Teaching of Higher Mathematics

Zhenzhen Li
School of Information Engineering, Jingdezhen University, Jingdezhen, 333000, Jiangxi, China
E-mail: lizz720@163.com

Keywords: higher mathematics, algebra, blended learning, multiple regression model (MRM), statistics

Recieved: March 1, 2024

Higher mathematics is a term used to describe complex mathematical ideas and subjects that go beyond the fundamentals of algebra, geometry, and number theory. Geometry, linear algebra, discrete mathematics, topology, and analysis are often covered. It entails creating fresh mathematical ideas and solving challenging issues by applying solid mathematical rationalization. Economics, statistics, mathematical modeling, and software for descriptive statistics are all areas covered by mathematics applications. Global concern is being raised by the falling number of students pursuing elevated amounts of mathematics. The underperformance and disinterest of students in mathematics may be attributed to a variety of issues. One cause of the drop is the knowledge gaps that arise when learners do not acquire or comprehend important mathematical ideas. It is essential to provide the best teaching strategy. Blended learning combines online and in-person instruction utilizing a range of tools and communication channels that are accessible to both students and instructors. In the setting of data processing and statistics, multiple regression analysis could serve as a helpful tool for teaching mathematics. Thus, we suggested using a multiple regression model (MRM) in blended higher mathematics instruction. Using performance measures and comparisons to existing methods, we assessed the efficacy of the suggested approach. The study results proved that MRM has provided an implementation cost of 45. According to the results, the proposed approach helps students learn mathematics in a more significant way.

Povzetek: Predstavljen je model za multiplo regresijo (MRM) v kombiniranem poučevanju višje matematike, ki združuje spletne in tradicionalne metode. Rezultati kažejo, da MRM izboljša učni proces, zapolni vrzeli v znanju in poveča motivacijo študentov.

1 Introduction

The concept of higher mathematics is used to describe more complex areas of mathematics that usually need a solid grounding in calculus, linear algebra, and other basic mathematical principles. Blended teaching is an approach to education that combines online resources with more conventional methods of instruction. Both the learner and the instructor should be in the same room for there to be a truly blended teachingexperience. Students may study at their speed using blended teaching. Learners may review previously covered ideas and materials at any time, and move on at their own pace. With a learning management system (LMS), learners may go at their speed while still having quick access to previously covered information. These fields of study are common in bachelor's degrees and higher graduate degrees in maths, physics, and engineering, and they find applications in disciplines including computing, encryption, finance, and data science [1]. In this method mixes face-to-face classroom instruction with distance education. That someone might, for instance, study the internet to have a general understanding of a subject, before attending a lecture in person to ask questions of specialists and hone their knowledge in more depth. Dealing with abstractions and arguments, as is common in higher maths, necessitates an excellent grasp of mathematical reasoning and issue abilities. Higher-level mathematics may be taught in a mixed format, giving students the advantages of both traditional classroom instruction and an internet study. Implementing a blended teaching program successfully requires setting goals and objectives that are explicit, quantifiable, agreed upon, practical, and efficient [2]. The foundations of the blended teaching paradigm center on the formulation of instructional materials and settings. Content training, individualized training, student autonomy, connections, discipline, and relevancy are just a few of the pillars. As the average score of the class using the blended teaching model is more than the minimal completeness requirement, it may be stated that blended teaching is more successful than traditional learning in increasing students' achievement of mathematical topics in tenth grade [3]. It is generally accepted that processors, graphical calculators, and the Web may be efficient mathematical tools, while the use of such technologies in mathematics teaching remains restricted in many settings. Although the term "blended teaching" is still not given a universally accepted meaning. Investigators of learning are becoming more interested in the potential of blended teaching environments for the instruction and study of mathematics [4]. Consider more about the "flipped classroom" technique: in this approach, students are given background information and readings from an outside category, and then return to use what they acquired in concern, group projects, and class discussions. The above strategy has the potential to shine in advanced mathematics, where drill and problem-solving are essential to mastery. Studies conducted at institutes have driven research into the implementation of flexible classrooms. The flipped classroom approach pushes students to use mathematical ideas to solve real-world issues, which helps them develop the critical reasoning and problem-solving abilities that are necessary for many different professions [5]. The analysis of simultaneous is used to analyzethe connection among aindependent measure and several dependent factors. Multiple regression models can be used to examine the correlation between various instructional methods, technological resources, or other elements and learner success in the setting of integrated learning. The efficacy of blended teaching models in contrast to conventional teaching can be assessed using multiple regression models. To determine that technology devices are most useful for improving student learning, multiple regression techniques could be utilized [6]. The application of technology as an educational tool aims to make learning more effective; among the pathways of learning effectiveness is the growth in students' reasoning ability; the anticipated potential is higher-order thinking skills, such as mathematical creativity. One of the elevated thinking abilities that students need to cultivate is the capacity to analyze creatively in mathematics. The amount of mathematics innovative reasoning ability is classified into four categories depending on that aspect, extremely creative, creative, pretty creative, and not creative [7].Researchers discovered negative attitudes in students including a complete absence of confidence in their ability to succeed across working hard, offense while beginning to experience rejection creates anxiety about giving incorrect responses is one indicator of resilience that was investigated but first showed up at the start of teaching, and when teachers asked students to answer questions and proceeded to the forefront of the class to explain to other companions, the majority of students, approximately 67percent, felt embarrassed to come to the front of the class. More than 50 percent of students, or around 18 students, remark that the issue is tough after studying the content and before working on the assignments [8]. The analysis year each student from a variety of programs revealed that online courses had a favorable impact on participation in tasks requiring mathematical reasoning, encouraging the utilization of internet learning in the context of mathematics instruction. Catch attention to the growing use of mobile technology in mathematics education, stressing that doing so has the impending to enhance performance by bringing mathematics instruction beyond the classroom and using specific benefits [9]. Multiple regression models could even, in general, be a useful tool for evaluating the efficacy of blended teaching in higher mathematics. Teachers may increase the quality of learning and improve learning performance by producing valuable teaching approaches, technological tools, and predictions of educational success [10]. The proposed multiple regression model (MRM) techniques for classifies in blended teaching of higher mathematics.

2 Related works

This section discusses the findings of several researchers, technical repots and research papers. The relevant literature is displayed in the table 1.

Table 1: Summary of related work

Reference	**Proposed**	**Result**	**Limitations**
[11]	Science technology and engineering mathematics (STEM) was success of elementary school students from poor socioeconomic regions was affected by conventional scientific education and integrated learning.	The research is a quasi-experimental investigation. Data from the student's mathematical resiliency survey along with the initial maths competency exam.	The impact of students' mathematical resilience after implementing blended teaching orientation learning models of behavioral conflict techniques.
[12]	The study described learning management system that is coupled with a powerful computational platform and automatic monitoring systems was used for the objective. It has been noted that students enrolled in biotechnology programs have developed more curiosity and enthusiasm in the field, which has led to improved competency.	The analyze results was regard to the problem-solving technique, the students were particularly able to identify its perks and how using the program had also been valuable to their success in their professional and academic lives.	Complexity, resource shortages, human mistake risk, and vulnerability to cyberattacks are some of the limitations. Monitoring systems must be balanced with processing power.
[13]	The article provided the summary of a quantitative comparative investigation on the benefit of blended teaching, especially the station rotation approach, on the arithmetic performance of 413 sixth-graders. Scores from the Measure of Academic Progress (MAP) and the State of Texas Assessments of Academic Readiness (STAAR) were utilized.	The results suggest that blended learning might be positively implemented in schools, especially for children that require greater educational support in a single academic year and are intellectually backward.	MAP examinations may not completely capture each student's progress and might not evaluate certain abilities or ideas in enough detail, thereby disregarding essential qualitative components of learning.
[14]	The research on the use of blended learning in the context of English as a second or foreign language. In the relatively young discipline of blended learning, traditional instructional methods are combined with online and remote learning.	Outcomes determined that blended learning is an excellent way to improve students' desire to learn the language, as well as their language proficiency.	The difficulties faced by language instructors while implementing blended learning are not well covered in the literature. Therefore, research is required to recognize and address such problems.
[15]	The article offered a specific example of blended education that combines conventional classes with internet community interactions and student-centered active learning environments. The objective is to improve a mathematics lesson's task design	Mathematical findings point to significant advantages in the coordination and utilization of various semiotic representation systems as they have evolved.	Mathematical coordination and application of many semiotic representation systems, which have developed throughout time, are limited by their intrinsic complexity, cognitive load, and interpretive difficulties, which hinder their

	that affect learners' mathematics achievement.		smooth integration and understanding in mathematical discourse.
[16]	Griffith Sciences Blended Learning Model was discussed in the section, along with the way it is utilized to integrate and manuscript blended concepts and structure in STEM education, the methodical support and instruction process was created, and the techniques employed to support the academic profession in teaching and learning.	A result of blended learning approach was created as a consequence of using technological advances to foster grassroots movements about its academic and professional trainees and create acquire for a shift in the blended learning technique.	Limited access to technology, possibilities for uneven student involvement, issues in sustaining human relationships, and difficulties in transferring hands-on activities to electronic submissions.
[17]	The article described the implementation of a blended learning method and the lessons learned to improve the arithmetic abilities and character of young adults in Bali's 8th grade. That improving teachers' ability to administer blended learning programs is crucial for improving students' mathematical aptitude.	Outcomes are to enhance teachers' ability to administer blended learning programs is crucial to raising students' morale and mathematical competency.	One disadvantage is that professors who are not experienced with blended learning techniques may object, which might prevent adoption from going effortlessly and affect students' learning produces and experiences.
[18]	The research used a quasi-experiment to analyze the impact of a tailored intervention technique on learners' achievement in the program and studying habits in a blended learning environment. The goal is to determine whatever information fusion technique uses their information to deliver the best achievements.	The results demonstrated that the students' academic performance improved as a result of adopting learning findings emphasize how crucial blended learning is to raising students' learning outcomes.	The absence of random assignment in quasi-experiments might undermine their internal validity. Limitations in controlling extraneous elements and confounding variables may erode the validity of correlations and generalizability.

3 Method

Higher mathematics is employed for a variety of purposes, including forecasting the behavior of intricate systems like weather patterns or financial markets, creating algorithms, developing encryption techniques for encrypted transmission, and simulating biological systems like genetics or epidemics. Higher mathematics study is crucial for pupils. As a result, we provided the multiple regression model for higher mathematics blended teaching. The dataset for the research is comprised of Chinese students. The proposed flow is depicted in figure 1.

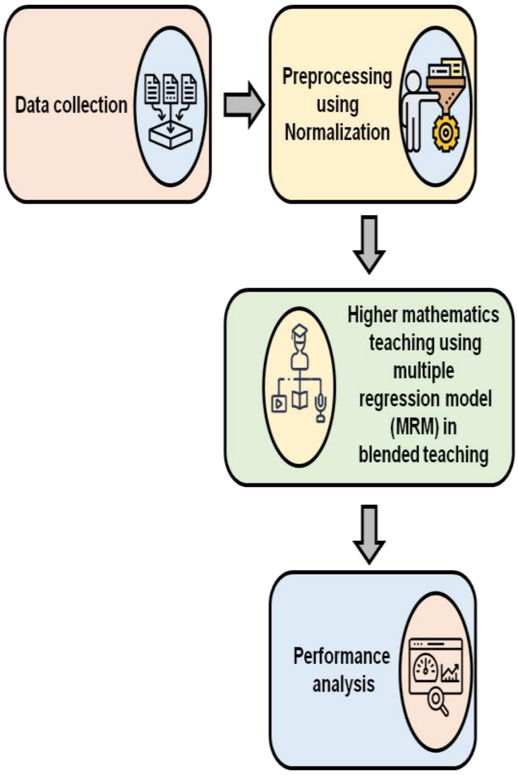

Data collection
Preprocessing using Normalization
Higher mathematics teaching using multiple regression model (MRM) in blended teaching
Performance analysis

Figure 1: Proposed architecture

3.1 Dataset

An amount of 408 students from the region of Guangxi are involved in this study. There are 317 female students and 91 male students, with a mean age of 14.3 years and 12.5 years, respectively. 22.3% of student is boys and 77.7% are girls, correspondingly. The 408 total students, 47.05% are junior high students and 52.94% are from primary schools. 11.4% of the 192 primary school pupils are male, and 88.6% are female. 14.81% of the 216 junior high school students are boys, and 85.19% are girls [19].

3.2 Data preprocessing using normalization

Data preprocessing describes the procedures and methods utilized to clean and convert raw data before being utilized for analysis or artificial learning algorithms. Preprocessing the information guarantees that it is reliable, comprehensive, and properly structured for analysis, making it a crucial stage in the data science process. In this study, we utilized min-max normalization. When a variable's range of entries is wide and the relative variances between those values matter, min-max normalization could be beneficial. Making ensuring that all variables have the same size is a typical practice in machine learning algorithms to enhance model performance. As a data-normalizing approach, min-max normalization scales numerical characteristics or variables to a predetermined range of values.

The method involves rescaling a variable's values such that they fall between 0 and 1, with 0 denoting the variable's smallest value and 1 denoting its greatest value. Equation 1 gives the calculation for min-max normalization.

$$Y_{normalized} = (Y - Y_{min}) / (Y_{max} - Y_{min}) \qquad (1)$$

where Y denotes the variable's initial value, Y_{min} denotes its lowest value, and Y_{max}denotes its maximum value.

3.3 Higher mathematics teaching using multiple regression model (MRM) in blended teaching

Mathematics could be taught well through blended learning, which mixes in-person and online instruction. The main feature of blended learning, which combines interactive and cooperative learning, was developed in this paper using topic materials, flipped classrooms, and student-centered instruction. Many online and offline exercises were created to fully achieve the connection between online and offline education. Self-paced training sessions including viewing films and taking quizzes are included, as well as interactive training tasks like peer assessment and class discourse. The tasks marked with the special icon were educational tasks that were carried out online using a website. The remaining exercises were mostly conducted offline and using blended learning. Before the session, most students learned on an online system. By viewing the videos, responding to the Q and A, or participating in the topic discussions with their students, they may get a taste of the course in preparation. Students are free to select the time and location for their online class exploration. The flexible teaching dominated in-class interactions. The instructor utilized a built-in mobile software named rainfall education tools to conduct random committee meetings and case analysis while teaching important and challenging learning points based on the student's educational circumstances. The center was used for the students. Students may engage in peer-to-peer discussions with instructors and their classmates, engage in activities, or communicate through on-screen remarks in a virtual classroom. Classroom instruction allowed for the smooth blending of online and offline education. Activities for learning after school were created to assist pupils in processing what they have learned. On the portal, multiple-choice questions (MCQs) and peer-review tasks were assigned to every unit of an instructional subject. The learning software arbitrarily produced the MCQ's choices and topics. In every lesson, there were two opportunities for students to finish their MCQ tasks. They have the option of responding twice or just once. Peer assessment aids students in comprehending and reconsolidating the information by assisting them in understanding the answers of their classmates. Students may examine the films, examine manuals, or refer to other pertinent resources as needed for these after-class projects. In this blended teaching MRM is incorporated to enhance the mathematics training.

In the environment of data research and statistics, a multiple regression model may be a helpful tool for math instruction. The efficacy of blended learning in advanced mathematics may be examined using MRM. The MRM attempts to evaluate how mixed learning affects mathematics produces by taking into account a number of variables, including instructional strategies, student involvement levels, and past educational achievement. The model may help teachers optimize their instructional strategies for enhanced academic performance in higher mathematics by revealing which components of blended learning contribute most substantially to student performance through statistical analysis. The following suggestions will help to teach mathematics using a multiple regression model. Before going on too many values, it starts with simple instances and introduces students to basic analytical concepts, including the connection between two variables. The fundamental ideas of the subject matter will be better understood by the students, as well as how they may be used to model connections between variables. It shows students practical applications of their studies, such as estimating the cost of a home based on location, size, and other criteria. Students will benefit from this as they learn about the real-world uses of arithmetic. It gives students the chance to use actual data and conduct numerous correlation assessments independently. Excel or other statistical tools may be used for this. It emphasizes how to analyze the findings. Instead of only computing the findings of the multivariate logistic analysis, it challenges students to concentrate on analyzing them. They will have a better understanding of how to forecast the future and reach conclusions using regression analysis as a result. It Determines the independent factors. The independent variables might include a variety of elements that could affect how well students learn arithmetic, including the student's previous math knowledge, the degree of the instruction, the resources on hand in the curriculum, and the student's degree of desire and engagement. Therefore, teaching math may benefit greatly from using multiple regression analysis, especially when it comes to documentation interpretation and statistics. MRM can be utilized to model connections between variables and create predictions, and by giving students practical knowledge and real-world applications, they can better understand higher mathematics.

4 Results and discussion

In this section, the proposed MRM effectiveness is evaluated. The assessment metrics used for evaluation such as understanding rate, pass rate, problem-solving, critical thinking, and implementation cost. The conventional techniques used for comparison are GeoGebra [20], Dynamic geometry software (DGS)[21], and Action learning and concept motivation (AL-CM) [22].

4.1 Understanding rate

An understanding rate is a unique ratio where the two terms are expressed in several units. Several techniques, including tests, quizzes, and other evaluations that gauge learners' comprehension of the subject matter, may be used to gauge the understanding rate in MRM learning. In order to ensure that students can understandhigher mathematics and achieve in their learning, educators may also regularly communicate with and provide comments to the learners, assess their progress, and modify the teaching techniques as necessary. The recommended technique has a higher level of understanding rate in mathwhenmeasured against the previously used methods. Figure 2 shows the understanding rate of the proposed and existing techniques. Table 2 shows the results of the understanding rate.

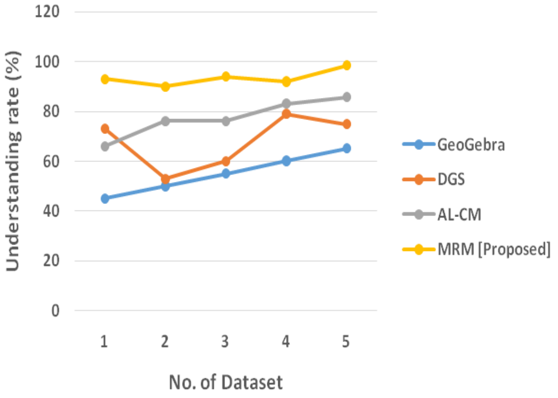


Figure 2: Understanding rate of the proposed and existing techniques

Table 2: Results of understanding rate

	Understanding rate (%)			
	GeoGebra	DGS	AL-CM	MRM [Proposed]
1	45	73	66	93
2	50	53	76	90
3	55	60	76	94
4	60	79	83	92
5	65	75	85.6	98.5

Table 3: Results of pass rate

Methods	Pass rate (%)
GeoGebra	55
DGS	66
AL-CM	75
MRM [Proposed]	97.9

4.2 Pass rate

The proportion of standardized tests that succeed is known as the pass rate. The proportion of learners that successfully finish the course or program with a passing grade is referred to as the pass rate in MRM learning. Sincepass rates differ between institutions and subjects, it is impossible to assigna certain pass rate to blended teaching. Figure 3 shows the pass rate of the proposed and existing techniques. Table 3 shows the results of the pass rate. In summary, effective implementation of MRM learning results in a high pass rate compared to the existing method.

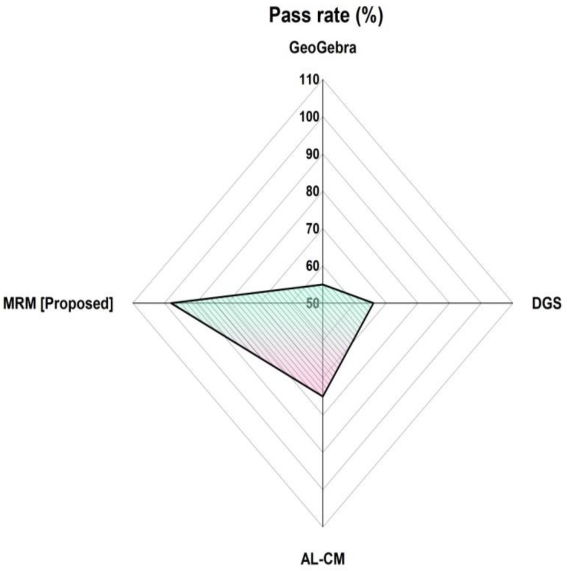


Figure 3: Pass rate of the proposed and existing techniques

4.3 Problem-solving

The expression problem-solving refers to skills that may be used todiscover the root of an issue and gives the best solutions. Problem-solving, being an integral aspect of the majority of actions, is the process of reaching a goal by overcoming barriers. Simple personal tasks and difficult challenges with blended learning are only a few of the problems solving. Information handling, reasoning, and imagination are the three skills that are the main elements of problem-solving. These three elements may be used to create problem-solving techniques that provide creative and successful responses to issues. Information handling, reasoning, and imagination are higher in the proposed method. Figure 4 shows the problem-solving of the proposed and existing techniques. Table 4 shows the results of problem-solving.

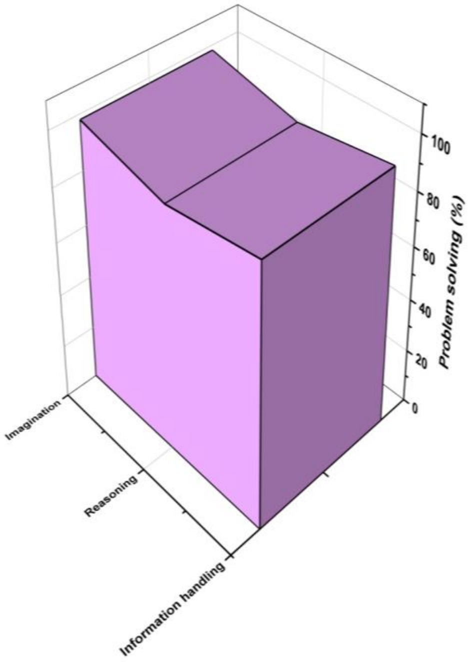


Figure 4: Problem-solving of the proposed and existing techniques

Table 4: Results of problem-solving

	Problem-solving (%)
Information handling	95
Reasoning	90
Imagination	98

4.4 Critical thinking

Questioning, analyzing, interpreting, and making a judgment about what one reads, hears, says, or writes are all elements of critical thinking. To become effective learners and hard workers, students must acquire the talent of critical thinking. A good setting for the development of critical thinking abilities is found in higher mathematicsbased on MRM learning. Figure 5 shows the critical thinking of the proposed and existing techniques. Table 5 shows the results of critical thinking.

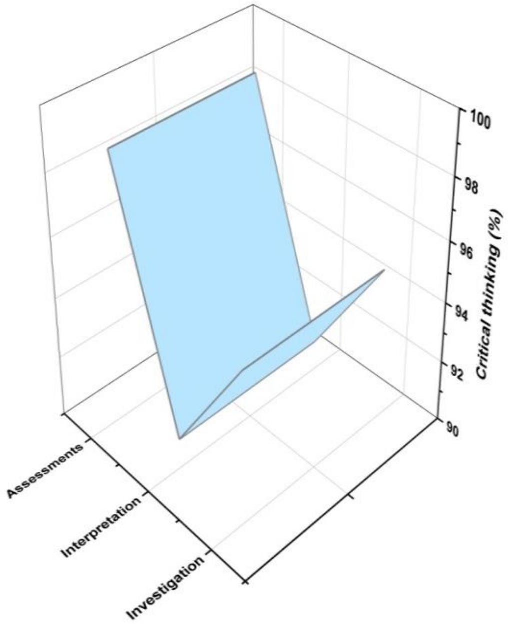


Figure 5: Critical thinking of the proposed and existing techniques

Table 5: Results of critical thinking

	Critical thinking (%)
Investigation	95
Interpretation	91
Assessments	98.8

4.5 Implementation cost

The cost incurred in developing and implementing a plan to implement one or more particular concrete proof techniques is known as implementation cost. The cost of implementing MRM teachingwill depend on the institution's particular requirements and objectives. These are some elements that may affect the price of implementing learningtechniques for higher mathematics. Figure 6 shows the implementation cost of the proposed and existing techniques. Table 6 shows the results of implementation cost. It proves that the suggested method uses less implementation cost.

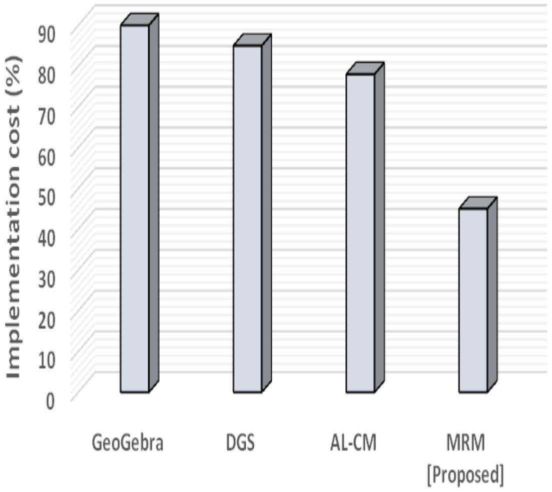


Figure 6: Implementation cost of the proposed and existing techniques

Table 6: Results of implementation cost

Methods	Implementation cost (%)
GeoGebra	90
DGS	85
AL-CM	78
MRM [Proposed]	45

5 Discussion

The drawback of using a GeoGebra is that it may be difficult for students to understand and manipulate multiple variables at once in the blended education. This could make it more difficult for students to understand fundamental mathematical concepts and increase cognitive load, which would hinder learning effectiveness. Several regression models into blended learning for higher-level mathematics are the possibility of increased interaction and visualization complexity. In a blended educational environment, students' comprehension and interaction with the regression recommendations may be hampered by DGS which is incomplete support for the complex data manipulation required. Higher education is that it might be too difficult for certain students, which will lower their interest in the subject matter. Since AL-CM emphasizes practical problem-solving and the application of mathematical ideas in practical situations, it might be better at producing desire. By analyzing these drawbacks, our suggested method MRM tends to overcome this issue and provide more efficient results

6 Conclusion

A multiple regression model (MRM) could be employed to examine the association between several independent factors and a dependent variable. MRM may be used to determine the variables that are most closely linked to student learning achievements when applied to the teaching of mathematics. MRM may be a very useful tool for engaging students and assisting them in understanding difficult mathematical ideas in higher mathematics. A method of education known as blended learning mixes online and in-person training in a manner that improves the learning process. Several disciplines, including physics, engineering, computer science, finance, and cryptography, utilize higher mathematics. It is crucial for simulating complicated processes and tackling issues that call for a better comprehension of mathematical ideas. Thus, we recommended the MRM in blended teaching of higher mathematics. The dataset used was composed of Chinese students. The data were normalized using Min-Max as part of the preprocessing. Evaluation and comparison with previously used methods are performed for the performance indicators understanding rate, pass rate, problem-solving, critical thinking, and implementation cost. The data showed that the suggested MRM provides students with an effective education in higher mathematics. A limitation of watching instructional videos during the coronavirus epidemic, according to parents, was that their children need their whole concentration. Higher education is provided more accessible and effectively when blended learning integrates conventional teaching strategies with internet resources. Technological obstacles, challenges in maintaining active engagement, and possible inequalities in resource accessibility are among the constraints. Specific educational processes, resource allocation, instructional approaches, and continuous improvement in course structure and execution are all made possible by the use of a multiple regression model in blended learning for higher mathematics. Such benefits subsequently translate into increased student engagement and achievement. The disadvantage of using the multiple regression model (MRM) in blended learning for higher education depends on assumptions regarding normality, linearity, and residual independence. Such presumptions could not always apply in complicated educational environments, which might jeopardize the model prediction of accuracy and generalizability. Regarding reliable results from studies, it is crucial to carefully evaluate the constraints and possible biases of MRM. In the future, innovative technology may be used with the suggested approach to maximize its efficacy.

MRM improvement: To increase the MRM's reliability and efficacy in estimating student outcomes, more optimization and refining of the model requires to be explored. Innovative technology integration wasthe prediction potential of the MRM will be enhanced by examining the integration of cutting-edge technology like natural language processing methods or ML algorithms. These tools may make it possible to analyze student data more thoroughly to gain a greater understanding of the trends and actions associated with learning.

References

[1] Attard, C. and Holmes, K., 2022. An exploration of teacher and student perceptions of blended learning in four secondary mathematics classrooms. *Mathematics Education Research Journal*, *34*(4), pp.719-740.

[2] Ustun, A.B. and Tracey, M.W., 2020. An effective way of designing blended learning: A three phase design-based research approach. Education and Information Technologies, 25(3), pp.1529-1552.

[3] Suma, K., Suwindra, I.N.P. and Sujanem, R., 2020. The effectiveness of blended learning in increasing prospective physics teacher students' learning motivation and problem-solving ability. JPI (JurnalPendidikan Indonesia), 9(3), pp.436-445.

[4] Galluzzi, F., Marchisio, M., Roman, F. and Sacchet, M., 2021, July. Mathematics in higher education: a transition from blended to online learning in pandemic times. In 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 84-92). *IEEE.*

[5] Muir, T., 2021. Self-determination theory and the flipped classroom: a case study of a senior secondary mathematics class. *Mathematics Education Research Journal*, 33, pp.569-587.

[6] Taghizadeh, M. and Hajhosseini, F., 2021. Investigating a blended learning environment: Contribution of attitude, interaction, and quality of teaching to satisfaction of graduate students of TEFL. *The Asia-Pacific Education Researcher*, 30, pp.459-469.

[7] Andriani, A., Dewi, I. and Sagala, P.N., 2019, March. Development of blended learning media using the mentimeter application to improve mathematics creative thinking skills. *In Journal of Physics: Conference Serie*s (Vol. 1188, No. 1, p. 012112). IOP Publishing.

[8] Fitri, S., Syahputra, E. and Syahputra, H., 2019. Blended learning rotation model of cognitive conflict strategy to improve mathematical resilience in high school students. *International Journal of Scientific & Technology Research*, 1(1), pp.80-87.

[9] Warren, L., Reilly, D., Herdan, A. and Lin, Y., 2021. Self-efficacy, performance and the role of blended learning. *Journal of Applied Research in Higher Education*, 13(1), pp.98-111.

[10] Müller, C. and Mildenberger, T., 2021. Facilitating flexible learning by replacing classroom time with an online learning environment: A systematic review of blended learning in higher education. *Educational Research Review*, 34, p.100394.

[11] Fitri, S. and Zahari, C.L., 2019, March. The implementation of blended learning to improve understanding of mathematics. In Journal of physics: Conference series (Vol. 1188, No. 1, p. 012109). IOP Publishing.

[12] Marchisio, M., Remogna, S., Roman, F. and Sacchet, M., 2020, July. Teaching mathematics in scientific bachelor degrees using a blended approach. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 190-195). IEEE.

[13] Fazal, M. and Bryant, M., 2019. Blended learning in middle school math: The question of effectiveness. *Journal of Online Learning Research*, 5(1), pp.49-64.

[14] Albiladi, W.S. and Alshareef, K.K., 2019. Blended learning in English teaching and learning: A review of the current literature. *Journal of Language Teaching and Research*, 10(2), pp.232-238.

[15] Capone, R., 2022. Blended learning and student-centered active learning environment: a case study with STEM undergraduate students. Canadian Journal of Science, *Mathematics and Technology Education*, 22(1), pp.210-236.

[16] Allan, C.N., Campbell, C. and Crough, J., 2019. Blended learning designs in STEM higher education. Blended Learn. Des. STEM High. Educ.

[17] Sudiarta, I.G.P. and Widana, I.W., 2019, October. Increasing mathematical proficiency and students character: lesson from the implementation of blended learning in junior high school in Bali. *In Journal of Physics: Conference Series (*Vol. 1317, No. 1, p. 012118). IOP Publishing.

[18] Nácher, M.J., Badenes-Ribera, L., Torrijos, C., Ballesteros, M.A. and Cebadera, E., 2021. The effectiveness of the GoKoan e-learning platform in improving university students' academic performance. Studies in Educational Evaluation, 70, p.101026.

[19] Wijaya, T.T., 2020. How chinese students learn mathematics during the coronavirus pandemic.

[20] Suryani, A.I. and Rofiki, I., 2020, February. The practicality of mathematics learning module on triangles using GeoGebra. In Journal of Physics: Conference Series (Vol. 1470, No. 1, p. 012079). IOP Publishing.

[21] Segal, R., Oxman, V. and Stupel, M., 2021. Using dynamic Geometry Software to enhance specialized content knowledge: Pre-service mathematics teachers' perceptions. *International Electronic Journal of Mathematics Education*, 16(3), p.em0647.

[22] Abramovich, S., Grinshpan, A.Z. and Milligan, D.L., 2019. Teaching mathematics throsugh concept motivation and action learning. *Education Research International*, 2019.

Cloud Computing Security: Assured Deletion

Sarmad Mahmood Ahmed, Baban Ahmed Mahmood
Department of Computer Science, College of Computer Science and Information Technology, Kirkuk University, Kirkuk, Iraq
E-mail: stcm22006@uokirkuk.edu.iq, babanmahmood@uokirkuk.edu.iq
*Corresponding Author

Student paper

Keywords: cloud service provider, cloud storage, assured deletion, policy revocation

Received: May 22, 2024

With the advent of cloud computing, many organizations, institutions, and individuals have chosen to store their data in the cloud as a way to compensate for limited local storage capabilities and reduce expenses. However, the process of uploading data to the cloud results in relinquishing control over it, leaving the data owner unaware of the details of its storage and location. Hence, ensuring data confidentiality and integrity has emerged as a critical concern, especially with regard to cloud employees and other potential attacks. A significant amount of investigation has been conducted on the security aspects of cloud computing, with a particular focus on permanent removals. Hence This study investigates the deployment of deletion assurance in cloud computing on two separate cloud platforms, employing strategies such as cryptographic key partitioning, cryptographic-based random writing, and link deletion within confirmed deletion scenarios. The characteristics of this approach include security safeguards like asymmetric encryption (specifically Elliptic Curve Cryptography (ECC)) and multiple hashing algorithms like hash 256 are used to fortify the security and confidentiality of data stored in the public cloud and the secrecy of files kept in the cloud is unaffected by invasions from the outside or from within an encrypted file prevents a threat from reading it and the owner may easily encrypt each file with a fresh key by managing and storing the keys using the key management center also authorized users are able to request this key thanks to secure access enforcement and In the event of assure deletion, the file is overwritten with random encryption and uploaded to the cloud, updating all copies while deleting all links and keys from the key management.

Povzetek: Študija se ukvarja z varnostjp oblačnega računalništva s poudarkom na zagotovitvi brisanja podatkov. Uporabljeni so algoritmi za kriptografsko brisanje, ECC šifriranje in delitev ključev za varno in zanesljivo odstranitev podatkov iz oblaka.

1 Introduction

Cloud computing facilitates effective and economical provision of resources to users upon request through the internet. The new operational framework of the organization promotes the utilization of the public cloud for online data retention and collaboration with external entities and institutions. Users are solely charged for the resources they utilize, resulting in diminished initial establishment and information technology infrastructure expenses. Cloud services empower individuals to stockpile and retrieve substantial volumes of data from a remote location, thereby diminishing the necessity for data maintenance and administration [1], [2], [3], [4]. With the use of commodity clusters, cloud computing has brought about a framework that makes it easier to assign data processing responsibilities to outside parties. By empowering data owners to trust third parties with their sensitive information, this approach reduces the computational and administrative costs involved in granting authorized users access to and manipulation of the data. When individuals and companies upload data to the cloud. You lose direct control over this data, so it is necessary to implement a security access policy for this data to ensure its security and integrity [1]. The purpose of secure deletion is to ensure that data is not used by unauthorized parties [5]. This goal is achieved by taking steps to prevent unauthorized access to your data after the deletion process is complete [6], [7]. Data must be encrypted before being sent to the cloud server. Therefore, most existing data deletion techniques rely on decryption key deletion to achieve the desired result of secure data deletion. Nevertheless, numerous challenges have arisen in this respect, such as the intricacy linked with key management, the protracted procedure of data destruction, and the restricted authority over data access [6]. Guaranteed deletion can protect the confidentiality of user data while also maintaining its accuracy [8]. All of these methods, meanwhile, fall short when it comes to the validation process used to evaluate the results of deleting data in the context of cloud computing. Thus, it is necessary to investigate the application of strong procedures for safe data removal as well as validation in the context of cloud computing [9]. There exist three primary methodologies. The most straightforward and highly efficient approach to achieve data eradication is via dissociation, or elimination. Eradication by superimposition: In order to supersede the original data within the cloud by updating it, the proprietor of the data engenders a data unit haphazardly that bears identical same name, classification, and magnitude as the data that necessitates erasure. The fundamental tenet underlying eradication by cryptology, or the annihilation of the decryption key, lies in the fact that the possessor of the data enciphers the dossier prior to externalization, and subsequently obliterates the decryption key of the enciphered dossier to render the encoded text infeasible for use [10], [11], [12], [13].

1.1 Contribution

In this paper, we put forward a hypothesis for a theoretical framework and plan for the encryption of information, the administration of cryptographic keys, and the preservation of keys. Additionally, we argue for the use of two separate cloud infrastructures to house both keys and documents. Furthermore, we propose the execution of a reliable method for data removal, employing a technique of overwriting using randomized encryption. Moreover, we suggest the elimination of keys and the revocation of the association policy.

The rest of the paper is organized as follows. Section 2 presents the most recent related work. The proposed method is presented in section 3. In section 4 Implementation and thorough analysis of the proposed method are presented. Section 5 concludes the paper.

2 Literature review

Researchers have suggested many deletion schemes based on policy revocation technology Overwriting, and other techniques. Below are some of the recent related works.

Tang et al. [14] designed a system for assured file deletion by revoking the file access policy. They depend on the encryption of data files and a data key to guarantee the privacy of files Initially, each data file is associated with a single policy or multiple policies through logical connections. In order to delete the file, the owner of the data revokes the policy associated with the file, and subsequently the key manager erases the private key rendering the data key irretrievable.

Yuchuan Luo et al. [15] proposed a new scheme for assured deletion called Permutation-based Assured Deletion Scheme (PADS) for the purpose of deleting data in cloud storage. This particular system involves the cloud generating data blocks in a random manner in order to overwrite the original data and ensure its recoverability. Additionally, it allows the data owner to confirm the results of the overwrite operation, ensuring successful deletion of data in the cloud.

Tyne et al. [8] A safe and effective ordered overwrite and erase scheme (SEAD-OO) is proposed. This method provides a multi-copy storage system for cloud data and adopts an ordered coverage method. The technology is composed of four basic components: Cloud Service Provider (CSP), Key Manager, Hyperledger Fabric, and Data Owners (including Authoritative Data Owner (ADO) and General Data Owner (GDO)).). ADO encrypts the file, uploads the ciphertext to the CSP, and passes the corresponding key to key management. ADO and GDO use the Diffie-Hellman protocol to establish session keys. Data accessibility is exclusively authorized based on the attributes of the data owner.

Zakaria and Mustapha [16] proposed a system to produce and store encryption keys locally rather than by a third party which was based on the Trusted Platform Module (TPM). Secure storage and cryptographic operations are offered by TPM, a hardware device often installed on the motherboard of a laptop or computer. By using the TPM chip's cryptographic features, the data is encrypted throughout the guaranteed deletion process, and a distinct key is created for it. Following the completion of the encryption procedure, only the encrypted data remains in the cloud storage with the original data being completely wiped. Because the encryption key is safely kept inside the TPM chip, FADETPM guaranteed deletion ensures that data cannot be accessed or recovered by unauthorized parties. This is what makes the technology unique.

Zhenjie Xie and colleagues [17] introduced a novel technique for ensuring the deletion of cloud data. By incorporating the XOR operation within a block cipher, the resulting cipher data becomes highly nonwearable, which presents an appealing approach to achieving assured deletion. The non recoverability of the original data subsequent to a deletion operation can be significantly enhanced by eliminating the key and certain encrypted data through the involvement of a trusted third party (TTP).

Wang and Luo [18] introduced a classification system for secure cloud data removal that is based on cryptographic methods. They examined this classification from two perspectives: one that involves a third-party key management centre and another that does not.

Joshi and Panchal [19] conducted a study on the requirements that need to be taken into account when ensuring assured data deletion in cases where the client does not have trust in the cloud server provider. These requirements include fine-grained deletion, cloud computation, availability of services, timeliness, complete deletion, and deletion acknowledgement. Furthermore, they presented existing techniques for ensuring assured deletion in the cloud, along with their limitations.

Table 1 highlights several strategies and mechanisms that have been utilized in the literature bringing out the most important features of each method.

Table 1: Comparison of various file assured deletion mechanisms based on common techniques

References	Assure deletion	Proof of deletion	Policy	Overwriting	KM	TTP
[8]	✓	✓	✓	✓	✗	✗
[9]	✓	✗	✗	✓	✗	✗
[14]	✓	✓	✓	✗	✓	✗
[15]	✓	✓	✗	✓	✓	✗
[16]	✗	✗	✗	✓	✗	✗
[17]	✗	✗	✗	✗	✗	✓

3 Methods and implementation

As we noted in Section 2, we propose a conjecture for a conceptual framework and blueprint for the encryption of data, the management of cryptographic keys, and the retention of said keys in a key custodian. Furthermore, we advocate for the utilization of two distinct cloud infrastructures to store both keys and files. Moreover, we put forward the implementation of a secure approach for data eradication, employing a process of overwriting utilizing randomized encryption. Additionally, we recommend the eradication of keys and the annulment of the association policy.

3.1 Preliminaries

In this section, we give four preliminaries encryption, Secure Hash Algorithm (SHA-256), Encryption Standard Algorithm (AES) and Rivest-Shamir-Adelman (RSA).

3.1.1. ECC encryption

Elliptic curve cryptography (ECC), a variant of the public key cryptosystem RSA, distinguishes itself by its enhanced adaptability and its provision of a desirable alternative for cryptographic algorithm researchers. In addition, the capacity of ECC to ensure a comparable level of security to RSA, despite using smaller key sizes, is noteworthy; for example, a 160-bit ECC key provides a level of security equivalent to that of a 1024-bit RSA key [20], [21].

3.1.2. Secure hash algorithm (SHA-256)

There are several options for cryptographic hash functions, including secure hashing algorithms. This algorithm is known as a "one-sided" cryptographic function, meaning that the original text cannot be recovered through decryption. The deterministic and consistent properties of a hash function ensure that it always produces the same hash result every time it processes a given message [22], [23].

3.1.3. Encryption standard algorithm (AES)

Known alternatively as private key cryptography, the Advanced Encryption Standard method (AES) is a cryptographic method based on symmetric cryptography principles. In this approach, the encryption and decryption operations are carried out using the same key. Several lengths of cryptographic keys are used to ensure safe data encryption. There are three different key size options: 128, 192, or 256 bits. Data is encrypted using AES in blocks of 128 bits each, functioning as a block cipher. As a result, it takes an input of the same length and outputs a 128-bit encrypted ciphertext[24].

3.1.4. Rivest-Shamir-Adleman (RSA)

the public key, which is used for encryption, is distributed to all users, while the private key, which is used for decryption, is kept secret. This is an example of an asymmetric cryptography algorithm. Usually, the keys have a length of 1024 or 2048 bits. Digital signatures and public key encryption are two applications for this method. The difficulty of factoring in huge prime numbers, which are used to produce the keys, is the foundation of its security[25], [26], [27].

3.2 System model

The principal constituents of a secure file deletion system, as delineated in Figure 1, are derived from the FADE-ECC model. The model comprises four distinct entities, namely the data owner, key manager, cloud service provider1, and cloud service provider2

3.2.1. Data owner - cloud users (DO)

The clients consist of both corporate entities and individual users. These clients engage in the rental of cloud services that are offered by the cloud service provider. These services entail a diverse selection of offerings, such as storage spaces, resources, infrastructure, and other related services. The purpose of availing these services is to leverage their economic and administrative merits, thereby alleviating the local burden associated with the utilization of these resources. Service consumers, in turn, establish an account with the cloud providers to make use of the most suitable cloud services available.

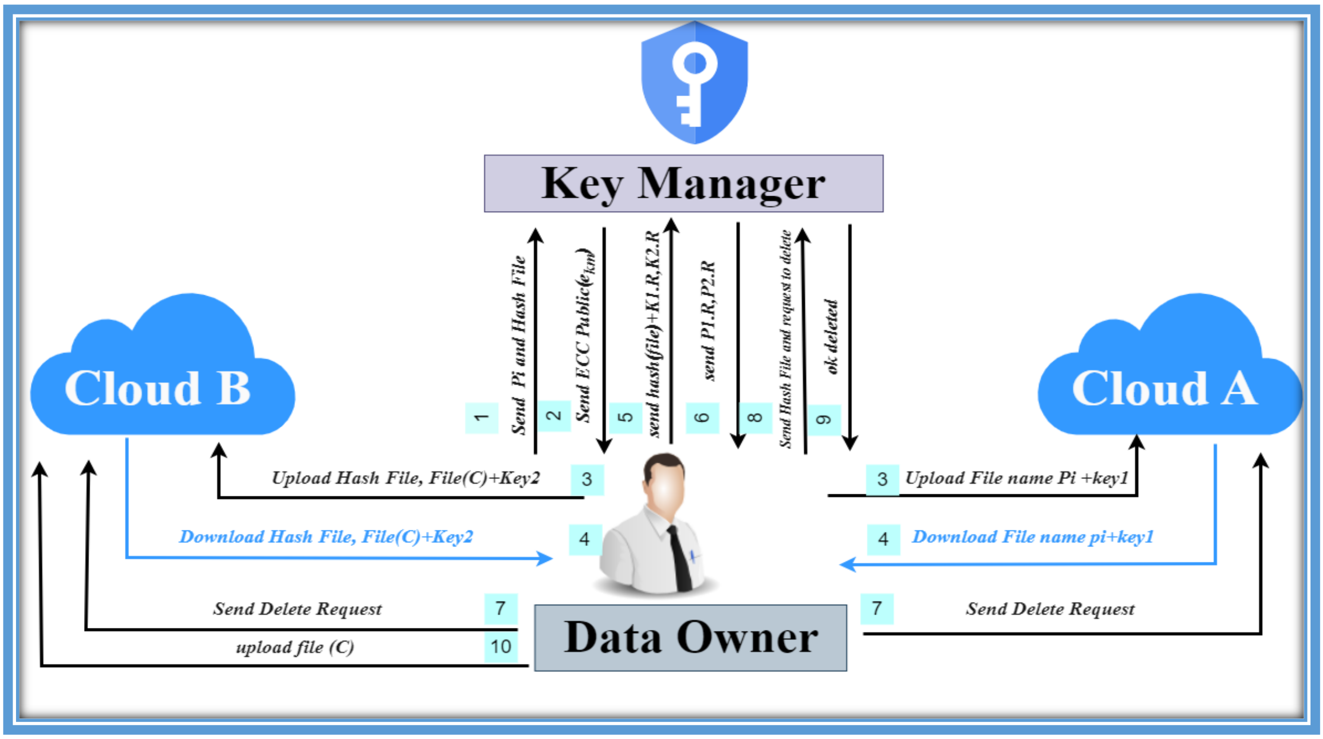

Key Manager
Cloud B
Cloud A
Data Owner
1
Send Pi and Hash File
2
Send ECC Public(e_{km})
5
send hash(file)+K1.R,K2.R
6
send P1.R,P2.R
8
Send Hash File and request to delete
9
ok deleted
Upload Hash File, File(C)+Key2
3
3
Upload File name Pi +key1
Download Hash File, File(C)+Key2
4
4
Download File name pi+key1
Send Delete Request
7
7
Send Delete Request
upload file (C)
10

Figure 1: System model

3.2.2. Key manager (KM)

The key manager is responsible for generating and managing the ECC private-public keys (control keys). Generates control keys based on the password hash value provided by the data owner, sends the public key to the data owner for use in encryption and keeps the private key for decryption. It deletes the private key when the data owner asks to delete it.

3.2.3. Cloud service providers 1 (CSP1) and 2 (CSP2)

These are two different businesses that provide elastic computing resources that are pay-as-you-go accessible to operations throughout the network.

3.3. System description

This section shows how the various components of the architecture work together and provides a high-level overview of its data activity, including uploads, downloads, deletions, and post-deletion verification.

3.3.1. Data encryption and uploading process

The public/private key (ei, di)' creation of the ECC is performed by the data owner (DO). The file is encrypted using the public key (ei), while the private key is split into two parts (d1f, d2f) based on the data owner's policy. Additionally, a SHA-256 hash of the file name is created and sent to the key manager. Key Manager (KM), on the other hand, creates two ECC type keys (ekm, dkm), maintains the private key (dkm) with a hash of the file name, and sends the public key (ekm) to the data owner. The keys that were divided are then encrypted using the public key of the key manager (KM) through ECC encryption. Afterwards, the encoded document and the initial encoded key are transferred to the initial cloud, while the checksum of the document is transferred with the second encoded key to the second cloud. Finally, all keys are erased, as illustrated in Figure 2

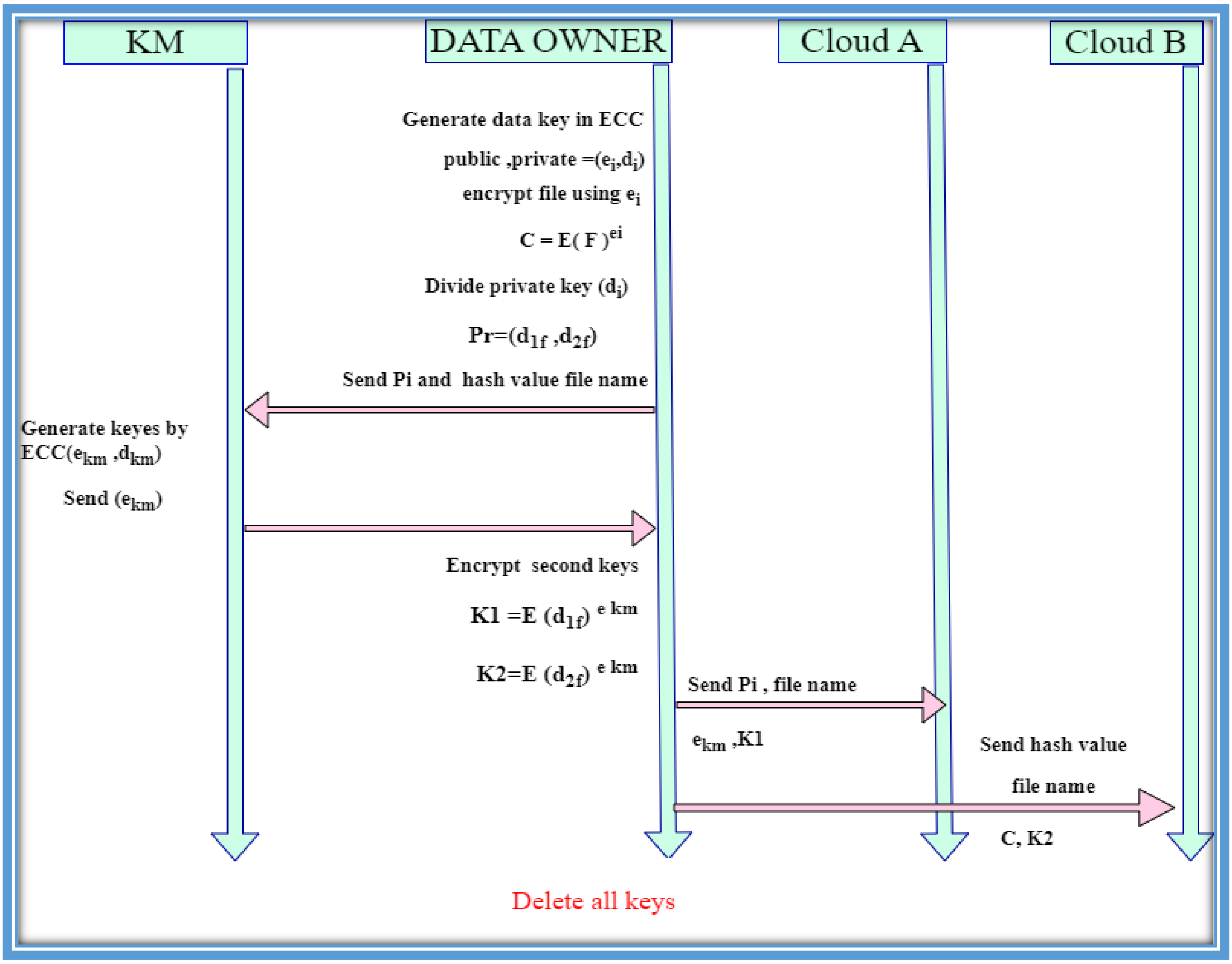


Figure 2: Data encryption and uploading process

3.3.2. File downloading and integrity checking process

The encrypted file (C) and the first encrypted key (K1) are downloaded by the data owner (DO) from the first cloud. Additionally, the data owner (DO) downloads the second encrypted key (K2) along with the file hash and the policy (pi) from the second cloud. Subsequently, a random value (R) is selected by the data owner (DO) and combined with the encrypted keys. These encrypted keys, along with the Hash-256 of the file name, are then transmitted to the key manager (KM).

Upon receipt, the key manager (KM) performs a comparison with the stored data. If a hash match is found, the sent keys are decrypted by the key manager (KM) using the ECC private key that is stored with him. The decrypted keys are then returned to the data owner. Following this, the random value (R) is removed and the keys (di) are collected based on the specified policy. Subsequently, the file is decrypted using ECC and finally opened. It is evident as depicted in the Figure 3.

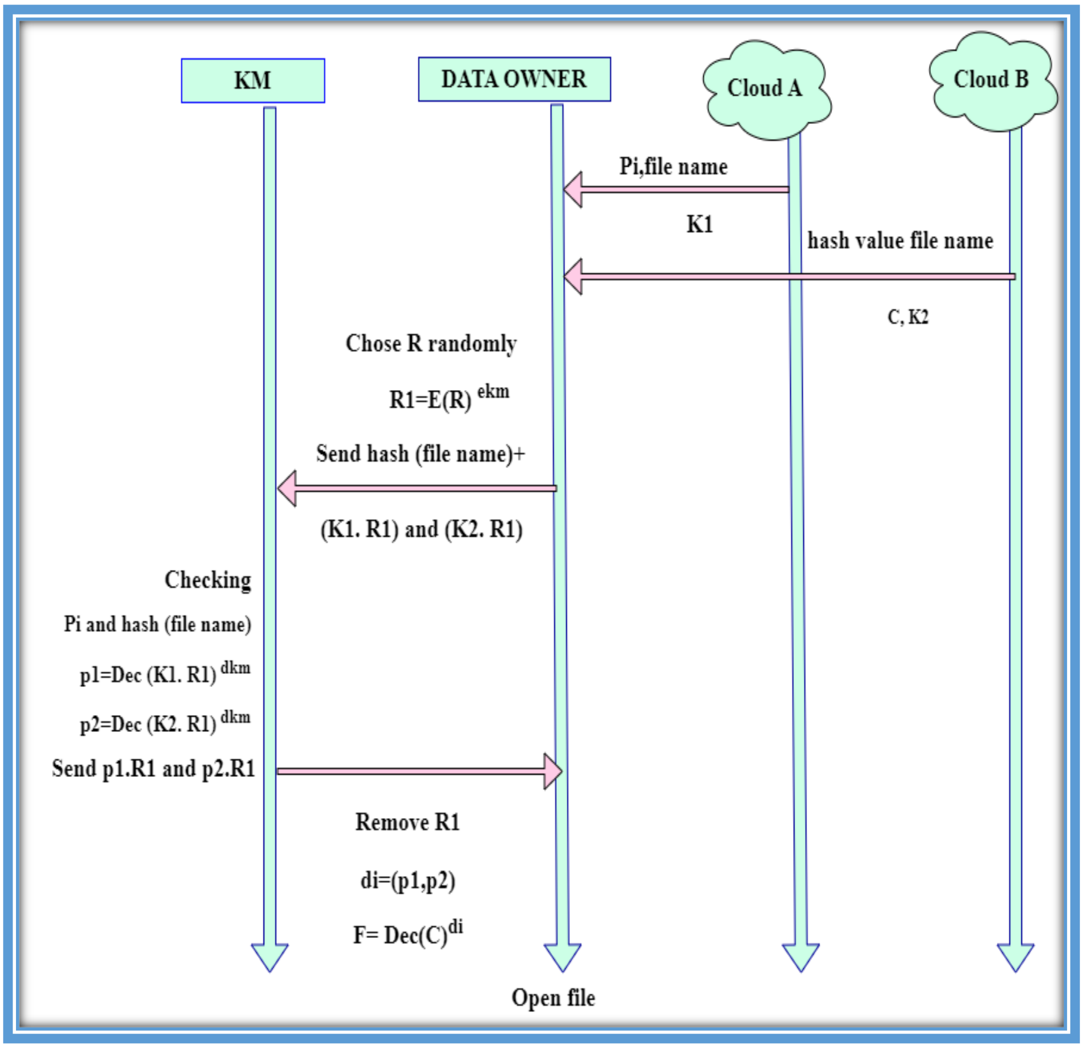


Figure 3: File downloading and integrity checking process

3.3.3 File deletion and verification process

In the process of confirmed deletion, the data owner (DO) employs a technique known as overwriting of the file. This involves encrypting the file a second time, utilizing a random public key of the ECC type without a private key. Simultaneously, the DO encrypts the file using the public key (PupR) and sends the Hash-256 of the file name to the key manager (KM). The KM is requested to erase the private key that they possess and subsequently re-uploads the file to the cloud, where the provider updates their copies. Following this, the DO asks the provider to erase the file. As a result, it can be ensured that the file stored in the cloud is rendered unusable since it is encrypted with a random encryption and lacks a first decryption key which has also been erased from key manager (KM), as portrayed in Figure 4.

3.4 Working environment

The mechanism known as FADE-ECC along with the FADE archetype were effectively implemented and executed on our Personal Computer as well as on the Firebase cloud. The process of execution entailed the utilization of a Python program in conjunction with a PyCharm editor on a Windows 10 Pro operating system. This particular OS is a 64-bit platform equipped with an Intel(R) Core (TM) i5-6300U processor that operates at a speed of 2.40GHz. Furthermore, the system was bolstered by a 16GB Random Access Memory (RAM) to support the execution of the tasks at hand. In order to gauge the performance of the application we proposed, a series of tests were conducted at various stages using files of different sizes, ranging from 1KB to 10 MB. The duration of the application's execution was carefully measured and then juxtaposed against that of the FADE mechanism at each stage. The primary objective of this evaluation was to assess the effectiveness of the implementation of our application. The interface of our implementation application is constructed based on the Tkinter module within the Python programming language. The functionality of our application commences with the input of credentials to grant access to the client for the use of the application. This process involves the design of a login system window, for entry. These fields require the authorized customer's user name and password for progression to the subsequent window of the application, which encompasses the stages involved in the implementation of our proposed design. Individuals lacking authorization to access the system or those without the requisite login credentials are unable to advance beyond the initial window of the simulated application. Consequently, our application serves as a barrier to prevent hackers and unauthorized users from infiltrating the system and gaining access to client data.

4 Analysis and experiment

The FADE-ECC mechanism and the FADE archetype were effectively executed on our Personal Computer and on the Firebase cloud. The execution involved the utilization of a Python program and a PyCharm editor on a Windows 10 Pro operating system, which is a 64-bit OS equipped with an Intel(R) Core (TM) i5-6300U processor operating at a velocity of 2.40GHz. Additionally, the system was supported by a 16GB Random Access Memory (RAM). To evaluate the performance of our proposed application, we conducted tests at each of the three stages using files of varying sizes, ranging from 1KB to 10 MB. The execution duration of our application was measured and compared with that of the FADE mechanism at each stage. This assessment aimed to evaluate the efficacy of our application implementation.

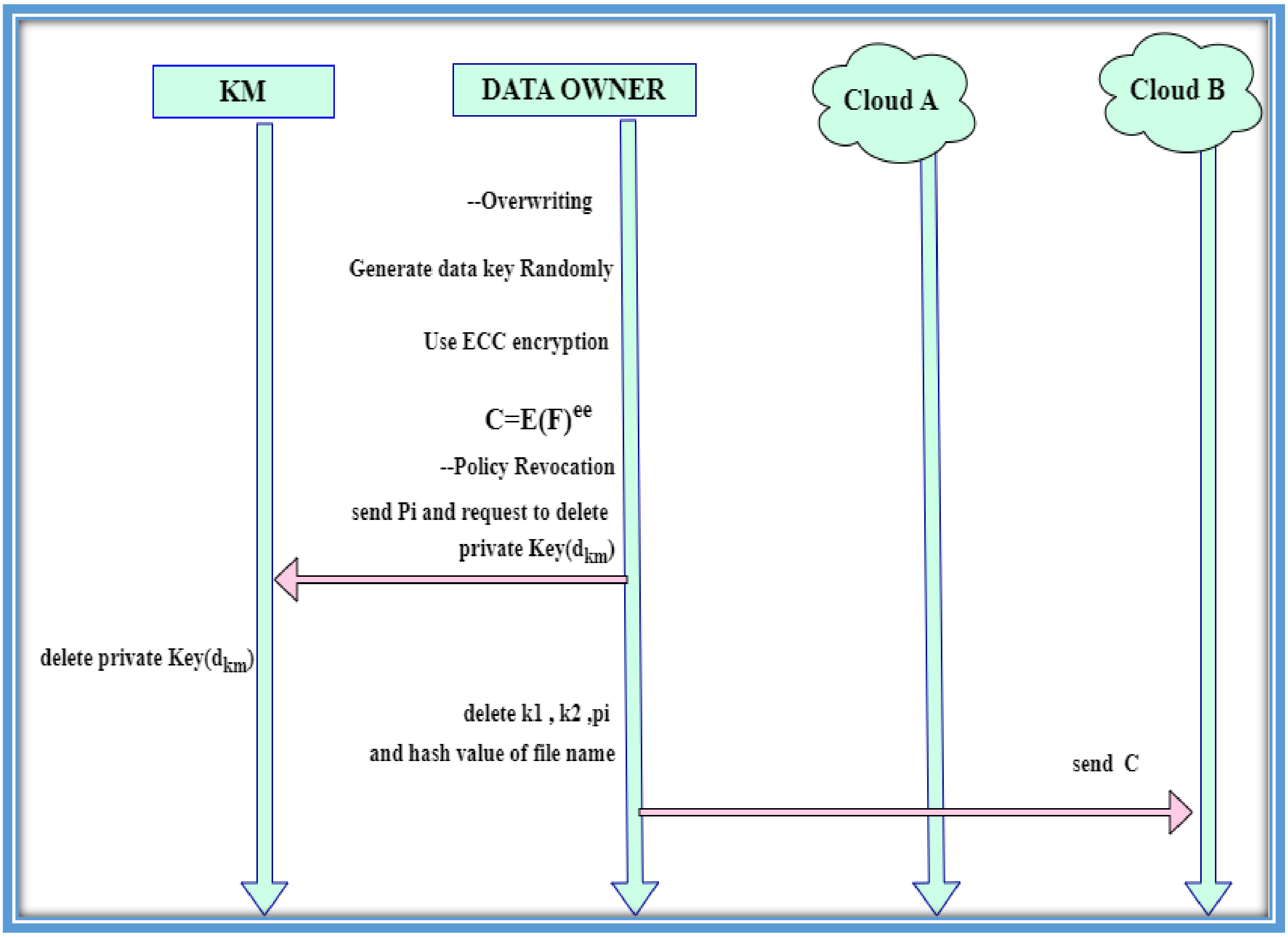


Figure 4: File deletion and verification process

4.1. File encryption and uploading stage

At this point, our system generates keys using ECC and choosing a policy, then encrypts the file, hashes the private key, and sends the hash with the file name to the key manager. The key manager also creates keys using ECC and sends the public key, then (encrypts) the keys and sends them to two clouds, while choosing the FADE model. At this stage, he chooses a policy associated with the file, then creates a key of type AES and sends the policy to the key manager to create two keys of type RSA and send the public key is sent to the file manager to encrypt the private key and send it to the cloud. The time has been calculated for the two models as shown in Tables 1 and 2 and compared in Figure 5.

Table 2: FADE-ECC Execution Time (Seconds)

File size	File encryption	Key manger	Key encryption	File upload	Total
1KB	0.000958	0.02295	0.02318	3.48912	7.32628
10KB	0.001351	0.030574	0.02336	4.80971	11.00565
100KB	0.000997	0.028453	0.03497	4.09435	11.75391
1M	0.004986	0.030191	0.02353	5.46235	12.93462
10M	0.030574	0.025616	0.019398	18.53329	34.32785

Table 3: FADE Execution Time (Seconds)

File size	File encryption	Key manger	Key encryption	File upload	Total
1KB	0.011967	3.647901	0.003008	2.255421	10.6916
10KB	0.016942	5.212392	0.002023	2.947587	12.9932
100KB	0.021533	5.574451	0.002091	4.155981	13.4686
1M	0.024003	7.75404	0.002093	5.233113	18.3231
10M	0.08769	9.14106	0.0019814	25.28561	41.9516

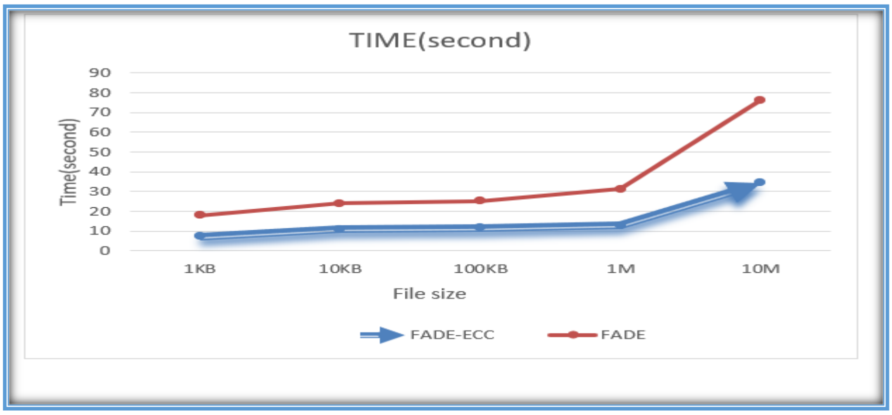


Figure 5: Comparison between the execution time of FADE and FADE-ECC

4.2 File downloading and decryption file

At this particular stage within our model, the individual who possesses the data proceeds to download both the file and the initial key from the first cloud. In addition to this, they also download the hash of the file and the second key from the second cloud. Following this, the individual in question selects a random value and combines it with the keys. Subsequently, they send this amalgamation, along with the hash of the file name, to the key manager. The key manager then proceeds to compare the hash with its own data and subsequently employs ECC to decrypt it using the provided keys. The key manager then transmits the decrypted keys back to the data owner, who then proceeds to gather the key and decrypt the file using ECC as well. In the FADE model, the data owner initially sends the key along with the random value to the key manager. The key manager then decrypts both the key and the random value using RSA, subsequently returning them to the data owner so that they may decrypt the file using AES. The time associated with both of these models is provided in Table 3 and 4, and is compared in Figure 6.

Table 4: FADE-ECC execution time (seconds)

File size	KM+ decryption keys	File decryption	Total
1KB	0.0132588	0.017235	0.0209423
10KB	0.0139636	0.017953	0.0329111
100KB	0.0150213	0.044862	0.0448623
1M	0.0130618	0.054103	0.0492492
10M	0.0120723	0.071856	0.0958753

Table 5: FADE execution time (seconds)

File size	KM+ decryption keys	File decryption	Total
1KB	0.019984	0.0301421	0.0321423
10KB	0.026381	0.0404043	0.0534001
100KB	0.029749	0.0571117	0.064261
1M	0.027643	0.0625601	0.077583
10M	0.023937	0.0940403	0.126302

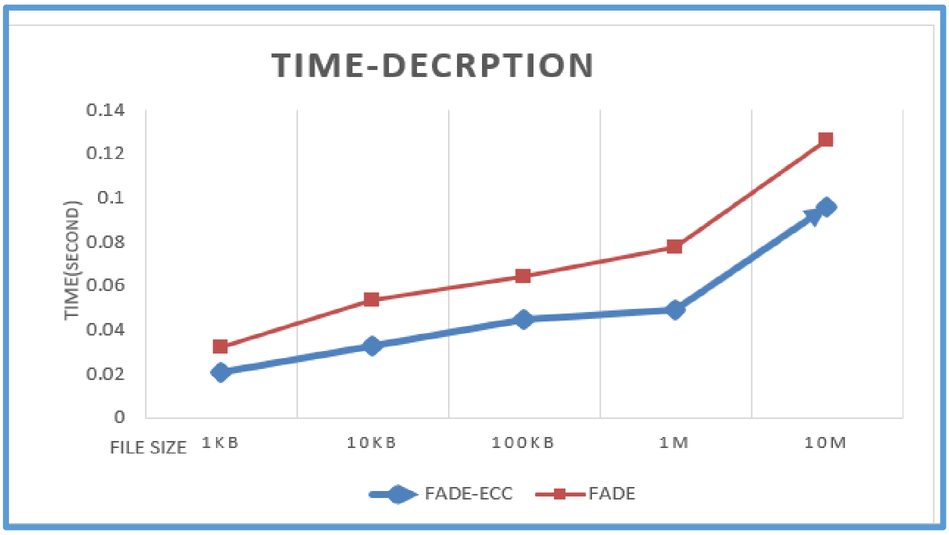


Figure 6: Comparison between the execution time of FADE and FADE-ECC

4.3 File deletion and verification phase

In our proposed model, this particular stage is characterized by the process of overwriting the file with a second layer of encryption using ECC. Additionally, this stage involves the removal of the initial encryption keys and the associated policy. In contrast, the FADE model entails the complete removal of the file from the cloud, deletion of the keys from the key manager, and elimination of the associated policy. This temporal progression of events is highlighted in table, specifically in the time frame of the two models, as depicted in Figure 7.

Table 7: Time of assure deleting FADE-ECC

File size	Key manger	Total
1KB	0.159873	5.967233
10KB	0.159855	11.57982
100KB	0.169753	16.68525
1M	0.168642	22.76039
10M	0.292135	51.80828

Table 8: Time of assure deleting FADE

File size	Key manger	Total
1KB	0.001999	9.391458
10KB	0.290616	16.231414
100KB	0.290593	16.542681
1M	0.265314	16.563211
10M	0.385641	18.749572

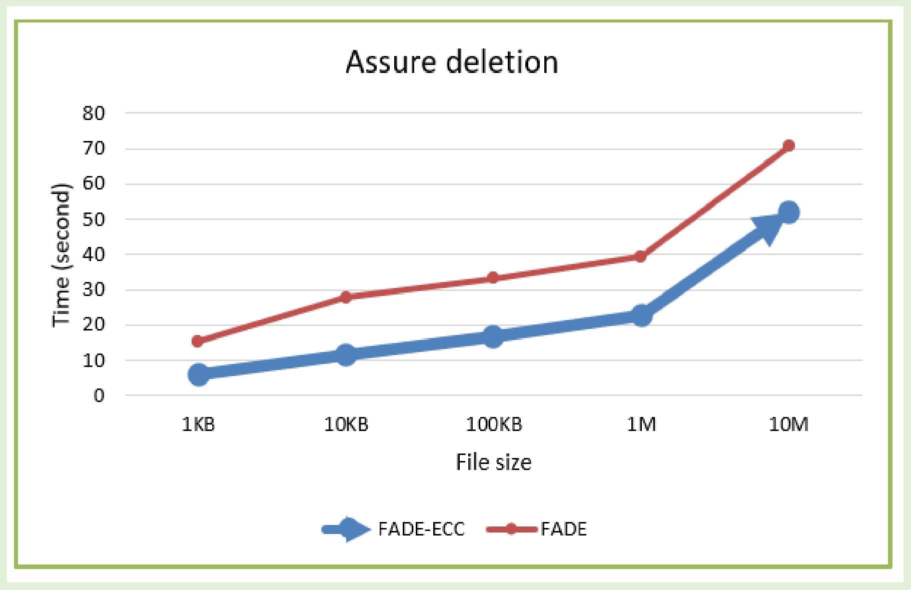


Figure 7: Comparison between the execution time of FADE and FADE-ECC

By implementing both our system and the FADE system on the same computer, the results we obtained showed the efficiency of our model in the encoding and decoding phases in terms of time. This is achieved through the use of ECC-type encryption, which features shorter keys and stronger security measures for both the data owner and the key manager. In contrast, FADE used AES encryption and RSA encryption for the key manager. From a security perspective, our model excelled in several aspects. we used two distinct clouds and used a file manager policy to split and distribute the keys. This ensures that sensitive files can only be accessed if all four parties cooperate, thus ensuring the integrity of our files even in the event of collusion between the service providers and the file manager, or in the event of an attack by an external party. Moreover, in the confirmed erasure phase, we used two confirmed erasure methods. The first method involved overwriting the file with random encryption and then uploading it to the cloud which would update all its copies. The second method involved all links and keys being deleted from the key manager. Thus, our approach provides enhanced security for sensitive data, unlike FADE, which relies on deleting links and keys within the file manager (KM).

5 Discussion

All the systems that have been put forward in previous years have exhibited a myriad of robust features and characteristics within their mechanisms aimed at the preservation of confidential data. However, there are certain aspects that may be deemed as weaknesses or areas that require improvement, such as the necessity to avoid certain practices and to adopt more potent or cutting-edge techniques in the realm of system design and operation. An exemplary model that served as a point of comparison with our own proposed system is FADE, which comprises three primary components: the data user, the file manager, and the cloud service provider. This system navigates through a sequence of three distinct stages: the initial phase involves the encryption of files followed by their upload to the cloud, then comes the stage of downloading files and subsequent decryption, and finally, the phase of confirmed scanning. The encryption of files in this system is facilitated by the application of AES encryption technology, while the encryption of the key is accomplished using RSA technology, in addition to the utilization of policies in conjunction with files. Contrarily, our proposed system incorporates four essential elements: the data owner, the file manager, cloud provider A, and cloud provider B, all of which are intricately interconnected in the context of the research endeavor. The indispensability of these elements is underscored by the fact that the research work would not be deemed complete in their absence. Similarly, our system progresses through three key stages, namely the encryption phase and subsequent uploading of files, the phase of downloading files from the cloud, decryption, and the stage of confirmed scanning. In the encryption of files and keys within our system, we opted for the utilization of ECC encryption technology owing to its reputation as the most robust form of encryption characterized by having the smallest key size and posing a formidable challenge in terms of code-breaking compared to other encryption methods. Additionally, we incorporated hashing techniques to ensure data integrity and prevent unauthorized tampering with the data. Upon an exhaustive evaluation of several systems within this domain, the efficacy of a system is contingent upon how well the encryption keys are handled and stored within the system infrastructure or the cloud environment. Notably, our proposed system stands out as one of the most robust systems due to our distinctive approach to key management and the integration of state-of-the-art technologies in the handling of sensitive files

6 Conclusion

Assured deletion has become challenging topic in the recent years. This paper introduces a suggested security system to tackle concerns in cloud computing by utilizing two separate cloud platforms, dividing encryption keys, utilizing overwriting methods, and unlinking connections through verified deletion. Hence, the proposed method fulfils both secure and integral file upload/download and assured deletion utilizing the above-mentioned techniques.

References

[1] S. Kaushik and C. Gandhi, “Capability Based Outsourced Data Access Control with Assured File Deletion and Efficient Revocation with Trust Factor in Cloud Computing,” International Journal of Cloud Applications and Computing, vol. 10, no. 1, pp. 64–84, 2020, doi: 10.4018/IJCAC.2020010105.

[2] A. Lec Ghassan Sabeeh Mahmood, “Data Security Protection in Cloud Computing by using Encryption ,” 2017. [Online]. Available: www.kujss.com

[3] A. Hameed and Ogr. Uye. O. Karan, “Cloud Data Storage Confidentiality Using Steganography and Visual Cryptography: A Review,” Journal of Education and Science, vol. 32, no. 4, pp. 60–70, Dec. 2023, doi: 10.33899/edusj.2023.140697.1370.

[4] H. Kumar, P. J. Soh, and M. A. Ismail, “Big Data Streaming Platforms: A Review,” Iraqi Journal for Computer Science and Mathematics, pp. 95–100, Apr. 2022, doi: 10.52866/ijcsm.2022.02.01.010.

[5] B. Li, Y. Fu, and K. Wang, “A Review on Cloud Data Assured Deletion,” in 2022 Global Conference on Robotics, Artificial Intelligence and Information Technology (GCRAIT), IEEE, Jul. 2022, pp. 451–457. doi: 10.1109/GCRAIT55928.2022.00101.

[6] J. Tian and Z. Wang, “Cloud data assured deletion scheme based on dynamic sliding window,” Peer Peer Netw Appl, vol. 15, no. 4, pp. 1817–1833, Jul. 2022, doi: 10.1007/s12083-022-01318-3.

[7] Y. Miao et al., “Privacy-Preserving Attribute-Based Keyword Search in Shared Multi-owner Setting,” IEEE Trans Dependable Secure Comput, vol. 18, no. 3, pp. 1080–1094, May 2021, doi: 10.1109/TDSC.2019.2897675.

[8] J. Tian and T. Zhang, “Secure and effective assured deletion scheme with orderly overwriting for cloud data,” Journal of Supercomputing, vol. 78, no. 7,

pp. 9326–9354, May 2022, doi: 10.1007/s11227-021-04297-z.

[9] J. Ma, M. Wang, J. Xiong, and Y. Hu, "CP-ABE-based secure and verifiable data deletion in cloud," Security and Communication Networks, vol. 2021, 2021, doi: 10.1155/2021/8855341.

[10] H. Ritzdorf, N. Karapanos, and S. Čapkun, "Assited deletion of releted content," in ACM International Conference Proceeding Series, Association for Computing Machinery, Dec. 2014, pp. 206–215. doi: 10.1145/2664243.2664287.

[11] K. Q. Aziz and B. A. Mahmood, "Assured data deletion in cloud computing: security analysis and requirements," Indonesian Journal of Electrical Engineering and Computer Science, vol. 28, no. 2, pp. 1174–1183, Nov. 2022, doi: 10.11591/ijeecs.v28.i2.pp1174-1183.

[12] J. Hao, J. Liu, W. Wu, F. Tang, and M. Xian, "Secure and Fine-Grained Self-Controlled Outsourced Data Deletion in Cloud-Based IoT," IEEE Internet Things J, vol. 7, no. 2, pp. 1140–1153, Feb. 2020, doi: 10.1109/JIOT.2019.2953082.

[13] S. M. Khudaier and B. A. Mahmood, "A Review of Assured Data Deletion Security Techniques in Cloud Storage," Iraqi Journal of Science, vol. 64, no. 5, pp. 2492–2511, 2023, doi: 10.24996/ijs.2023.64.5.33.

[14] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "FADE: Secure Overlay Cloud Storage with File Assured Deletion."

[15] Y. Luo, M. Xu, S. Fu, and D. Wang, "Enabling assured deletion in the cloud storage by overwriting," in SCC 2016 - Proceedings of the 4th ACM International Workshop on Security in Cloud Computing, Co-located with Asia CCS 2016, Association for Computing Machinery, Inc, May 2016, pp. 17–23. doi: 10.1145/2898445.2898447.

[16] Z. Igarramen and M. Hedabou, "FADETPM: Novel Approach of File Assured Deletion Based on Trusted Platform Module," 2019, pp. 49–59. doi: 10.1007/978-3-319-97719-5_4.

[17] Z. Xie, W. Fu, J. Xu, and T. Zhu, "Assured Deletion: A Scheme Based on Strong Nonseparability," J Sens, vol. 2022, 2022, doi: 10.1155/2022/9691724.

[18] G. Wang and Y. Luo, "A Review on Assured Deletion of Cloud Data Based on Cryptography," Procedia Comput Sci, vol. 187, pp. 580–585, 2021, doi: 10.1016/j.procs.2021.04.111.

[19] S. B. Joshi and S. D. Panchal, "A Survey on Assured Data Deletion in Cloud Storage," International Journal of Computer Sciences and Engineering, vol. 7, no. 6, pp. 548–553, Jun. 2019, doi: 10.26438/ijcse/v7i6.548553.

[20] S. Kumar Verma and D. Ojha, "A Discussion on Elliptic Curve Cryptography and Its Applications," 2012. [Online]. Available: www.IJCSI.org

[21] M. Ma, "Comparison between RSA and ECC," in 2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), IEEE, Oct. 2021, pp. 642–645. doi: 10.1109/AINIT54228.2021.00129.

[22] R. Martino and A. Cilardo, "A Configurable Implementation of the SHA-256 Hash Function," 2020, pp. 558–567. doi: 10.1007/978-3-030-33509-0_52.

[23] S. R. Prasanna and B. S. Premananda, "Performance Analysis of MD5 and SHA-256 Algorithms to Maintain Data Integrity," in 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), IEEE, Aug. 2021, pp. 246–250. doi: 10.1109/RTEICT52294.2021.9573660.

[24] R. Saha, G. Geetha, G. Kumar, and T. Kim, "RK-AES: An Improved Version of AES Using a New Key Generation Process with Random Keys," Security and Communication Networks, vol. 2018, pp. 1–11, Nov. 2018, doi: 10.1155/2018/9802475.

[25] A. H. Thiziers, H. Cisse, J. T., and B. Michel, "Enhanced, Modified and Secured RSA Cryptosystem based on n Prime Numbers and Offline Storage for Medical Data Transmission via Mobile Phone," International Journal of Advanced Computer Science and Applications, vol. 10, no. 10, 2019, doi: 10.14569/IJACSA.2019.0101050.

[26] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion," IEEE Trans Dependable Secure Comput, vol. 9, no. 6, pp. 903–916, Nov. 2012, doi: 10.1109/TDSC.2012.49.

[27] Shahla Uthman, "An Improved RSA based on Double Even Magic Square of order 32," Kirkuk University Journal /Scientific Studies (KUJSS), vol. 12, no. 1992 – 0849, 2017.

Tropical Cyclone Intensity Estimation by Feature Extraction Techniques Using Satellite Imagery

Chinmoy Kar
Techno International New town, Kolkata, India
E-mail: chinmoy.kar@tint.edu.in

Thesis Summary

Keywords: tropical cyclone, intensity estimation, feature extraction, machine learning classifiers

Received: July 14, 2024

This article provides an overview of the author's doctoral dissertation entitled "Intensity Estimation of Tropical Cyclones by Feature Extraction Techniques". The dissertation introduces feature extraction techniques applied to satellite images of tropical cyclones to extract meaningful features. Additionally, the dissertation explores the use of various machine learning classifiers to determine the intensity class of a tropical cyclone based on these features.

Povzetek: Predstavljene so metode za ocenjevanje intenzivnosti tropskih ciklonov z uporabo ekstrakcije značilnosti iz satelitskih slik. Uporabljene so metode strojnega učenja za razvrščanje slik ciklonov glede na intenziteto, pri čemer sta se večplastni perceptron in naključni gozdovi izkazala za najučinkovitejša.

1 Introduction

Tropical cyclones (TCs) are natural hazards that cause massive damage near the landfall area. Early warning through TC intensity assessment helps to mitigate this damage. The wind field analysis and pattern matching-based methods. Wind field attributes are combined with best track data to determine the intensity of a TC. Pattern matching-based methods, such as the Dvorak Technique [1] and Advance Dvorak Technique [2], involve analyzing TC images to find T numbers based on the cloud pattern, which are then used to assess the TC intensity and its impact. In addition to these methods, numerical weather prediction (NWP) models, which rely on geophysical properties like air pressure, wind speed, and sea surface temperature, are used for TC intensity estimation. Image processing-based approaches are also utilized to estimate TC intensity based on the cloud pattern, curved bands, and shear structure of the cloud. In recent years, limited applications of deep learning algorithms have been used for TC intensity estimation [3].

The traditional methods for estimating cyclone intensity rely on geophysical properties and require human intervention for analysis. The main goal of this doctoral dissertation was to investigate various feature extraction methods and a fusion technique for cyclone intensity estimation. Furthermore, this research focuses on studying cyclones occurring over the North Indian Ocean (NIO) basin, aiming to develop image recognition-based techniques that can mitigate the damage caused by cyclones.

2 Methods

The subsequent subsections provide a summary of the authors' contributions.

2.1 Development of unique feature vector from satellite images [8, 11]

A method has been proposed to create a unique feature vector (UFV) from TC images by using rotation-size invariant patterns. The main features of an image, such as the center of gravity [11], density, and eccentricity, are utilized to identify rotation and size invariant patterns. Any method that estimates intensity based on TC images heavily relies on rotation-size invariant patterns and region of interest (RoI) images.

2.2 TC image classification using UFV and machine learning classifiers [5,7]

The intensity of tropical cyclones (TC) is estimated using the UFV and machine learning techniques for classification. Various machine learning classifiers are utilized to categorize TC images into seven intensity levels. The Multilayer Perceptron and Random Forest classifiers showed superior performance compared to other classifiers.

2.3 TC Intensity estimation by local binary pattern (LBP) and its variant [4,10]

The Multilayer Multi-block LBP (MMLBP) method is an extension of the Completed LBP (CLBP). It extracts important features in a step-by-step manner, organizing the input image into fixed-size blocks (3×3 pixels) to create center pixels using the CLBP. These central pixels serve as the basis for the next layer, which is further divided into 3×3 blocks to generate central pixels. The central pixels from each layer are then collected and combined into a feature vector.

2.4 Fusion based technique for intensity estimation [6]

A feature fusion and machine learning-based classification techniques to estimate the intensity of Tropical Cyclones (TC). Geometric features, Haralick features, and Multi-Block LBP features from TC images.

3 Discussion

The doctoral research involved analyzing Infrared (IR), Visible (VIS), and Water Vapor (WV) TC images of seven classes: Super Cyclone Storm, Extremely Severe Cyclonic Storm, Very Severe Cyclonic Storm, Severe Cyclonic Storm, Cyclonic Storm, Deep Depression, and Depression over the North Indian Ocean basin [9]. It was found that the UFV-TC and Random Forest (RF) classifier had an accuracy of 86.66%, which is higher compared to other classifiers. The main advantage of RF classifiers is their effectiveness in multi-class classification. By combining a group of weak individual learners into one strong learner using the ensemble model classifier, the highest accuracy can be achieved. The MMLBP and Random Forest classifier achieved a classification accuracy of 84.66% for feature extraction and feature vector creation from an image. MMLBP generated 819 features, which was then reduced by up to 97.4% using the correlation-based Feature Subset Selection approach. A Min-Max normalization-based fusion method was applied to combine GLCM, MB-LBP, and geometric features. Furthermore, a small set of features was selected from a large feature vector by reducing the original feature vector by 93% using the correlation-based feature subset selection method. The fused feature vector was able to classify TC images of various classes with 93.5% accuracy.

4 Conclusions

The thesis focuses on feature extraction techniques for tropical cyclone (TC) images over the North Indian Ocean (NIO) basin. The rotation-size invariant unique feature vector generation is explored, followed by a classification task to categorize TC images into various classes. Local binary patterns from cloud images are extracted using LBP and a modified LBP algorithm named Multi Block Multi-Layer LBP (MMLBP). Haralick features from TC images are also extracted, and a feature fusion technique is proposed to fuse Haralick features, Multi Block LBP (MB LBP) features, and geometric features. The proposed feature extraction techniques are applied to IR, VIS, and WV satellite images for performance analysis on various satellite images. The study also collects and checks performance towards intensity estimation over the NIO basin. The RF classifier and MMLBP achieve the highest classification accuracy compared to other tree-based classifiers.

The thesis presents several limitations, including the lack of standard TC databases for the NIO basin, difficulty in gathering accurate TC images for cyclonic seasons, and a lack of comparative analysis between similar models due to limited cyclone image feature extraction-based intensity estimation techniques. The study was tested using NIO basin images and a limited dataset of other basins. The Future work includes addressing the need for labelled TC images, using multispectral images for intensity estimation, testing the model on Pacific and Atlantic basins, and incorporating additional techniques for extracting and combining features to improve the TC estimation model for the NIO basin.

Acknowledgment

I would like to express my gratitude to my supervisor, Prof. (Dr.) Sreeparna Banerjee, Maulana Abul Kalam Azad University of Technology, West Bengal, India.

References

[1] Dvorak, V. F., Tropical cyclone intensity analysis using satellite data. In NOAA Technical Report NESDIS, 11, US Department of Commerce, National Oceanic and Atmospheric Administration, National Environmental Satellite, Data, and Information Service: Washington, DC, USA, 1–47, (1984).

[2] Olander, T. L., & Velden, C. S., The advanced Dvorak technique (ADT) for estimating tropical cyclone intensity: Update and new capabilities. Weather and Forecasting, 34(4), 905-922, (2019).

[3] Pradhan, R., Aygun, R. S., Maskey, M., Ramachandran, R., & Cecil, D. J. (2017). Tropical cyclone intensity estimation using a deep convolutional neural network. IEEE Transactions on Image Processing, 27(2), 692-702.

[4] Kar, C., Banerjee, S., Intensity prediction of tropical cyclone using multilayer multi-block local binary pattern and tree-based classifiers over North Indian Ocean, Computers & Geosciences, 154, 1-14, (2021).

[5] Kar, C., Banerjee, S., Tropical cyclone intensity classification from infrared images of clouds over Bay of Bengal and Arabian Sea using machine learning classifiers. Arabian Journal Geosciences 14, 683 (2021).

[6] Kar, C., Banerjee, S., Tropical Cyclones Intensity Estimation by Feature Fusion and Random Forest Classifier Using Satellite Images. Journal of Indian Society Remote Sensing 50, 689–700 (2022).

[7] Kar, C., Kumar, A., Banerjee, S., Tropical cyclone intensity detection by geometric features of cyclone images and multilayer perceptron. SN Applied Sciences 1, 1099 (2019).

[8] Kar, C., Banerjee, S., An image processing approach for intensity detection of tropical cyclone using feature vector analysis. International Journal of Image and Data Fusion, 9(4) 338-348 (2018).

[9] Kar, C., Banerjee, S., Intensity Estimation of Tropical Cyclone using Different Satellite Imagery and Random Forest Classifier, 2022 IEEE Region 10 Symposium Mumbai, India, 2022, pp. 1-5.

[10] Kar, C., Banerjee, S. Tropical Cyclones Classification from Satellite Images Using Blocked Local Binary Pattern and Histogram Analysis. In Advances in Intelligent Systems and Computing, vol 1248, (2021).

[11] Kar, C., Kumar, A., Konar, D., and Banerjee, S. Automatic Region of Interest Detection of Tropical Cyclone Image by Center of Gravity and Distance Metrics, 2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, 2019, pp. 141-145.

Contact-Free Physiological Monitoring of Cardiorespiratory States Using Radar and Optical Sensors

Gašper Slapničar
Department of Intelligent Systems, Jožef Stefan Insitute, Ljubljana, Slovenia
E-mail: gasper.slapnicar@ijs.si

Thesis Summary

Keywords: contact-free, physiological monitoring, radar, camera, photoplethysmogram, blood pressure, hemodynamics

Received: September 25, 2024

The paper summarizes a Doctoral Thesis that focuses on two new approaches for unobtrusive contact-free monitoring of cardiorespiratory and hemodynamic states. First approach is based on radar signals and proposes a novel branched neural network architecture for classification of hemodynamic scenarios. The second is based on RGB camera signals and proposes multi-wavelength depth-dependant photoplethysmogram reconstruction, allowing for single-site pulse transit time measurement and blood pressure estimation using a consumer camera.

Povzetek: Članek povzema doktorsko dizertacijo, ki se osredotoča na brezstično zaznavanje fizioloških signalov z uporabo radarja in RGB kamere.

1 Introduction

Omnipresence of sensor-equipped devices spurred rapid development of e-health and m-health applications in the past decades. Despite their wide-spread adoption in the form of wearables [1], such devices are ultimately not a universal or ideal solution for regular health monitoring due to their reliance on battery, requiring skin contact and general obtrusive nature. Ideally, the need for direct user-device interaction should be completely removed in the paradigm of ubiquitous and pervasive computing, which can be achieved using contact-free sensors such as radars and cameras that monitor different parts of electromagnetic (EM) spectrum [2]. These devices can be used to monitor different physiological parameters unobtrusively, making them feasible for subjects who cannot use wearables (e.g., neonates, burn victims, elderly with dementia) [3].

2 Radar-based classification of hemodynamic scenarios

We initially explored the potential of radio-frequency part of the EM spectrum, measured by radars, for detection of complex hemodynamic states. These are expressed via several physiological parameters, including respiration. Radars allow for measurement of periodic thoracic expansion and contraction even in challenging conditions, such as night time and occlusion, making them ideal for sleep monitoring. We proposed a novel branched neural network architecture that can take a different number and type of input signals [4]. Two types of layers were primarily investigated, namely 1D convolutional networks (1D CNN) and fully connected networks (Dense ANN). Several input options were checked in terms of window length, modality (contact, radar, or fusion) and data type (temporal, frequency representation via FFT, or fusion).

We showed that we can detect five different hemodynamic states available in a public dataset [5] (Apnea, Valsalva, Tilting table ascent, Tilting table descent, Resting) with up to 0.83 accuracy and F1 score when using only contact-free radar signals as input. These results were only 4-5% behind traditional contact sensors, as shown in Table 1, confirming feasibility of radar-based physiological monitoring.

3 Camera-based MW PTT measurement and BP estimation

In the second part, we investigated the feasibility of using the visible part of the EM spectrum, specifically the feasibility of a modified consumer RGB camera for multi-wavelength (MW) pulse transit time (PTT) measurement between different skin layers [6]. Different wavelengths penetrate to different depths and allow for depth-specific photoplethysmogram (PPG) reconstruction, as shown in Figure 1.

These PPGs can be used for MW PTT computation and subsequent blood pressure (BP) estimation. We found that algorithmic channel separation of mentioned PPGs is mandatory due to the imperfect nature of image sensor design, which causes spectral overlap between PPGs from different depths. We thus developed several algorithms that

Table 1: Accuracy and F1 score (Acc. / F1) for the investigated Dense and 1D CNN networks at different window lengths, input modalities, and input data types, always using the best-performing set of hyperparameters. Best results for each network architecture are bolded, and the overall best results are highlighted in green.

Window	**Network**	**Modality**	**Temp. Data**	**Freq. Data**	**Temp. + Freq.**
5 s	*Dense ANN*	Contact	0.69 / 0.68	0.68 / 0.66	0.69 / 0.66
		Radar	0.64 / 0.61	0.72 / 0.71	0.72 / 0.72
		Fusion	0.68 / 0.68	0.66 / 0.65	0.70 / 0.70
	1D CNN	Contact	0.65 / 0.65	0.67 / 0.65	0.70 / 0.69
		Radar	0.62 / 0.60	0.61 / 0.60	0.62 / 0.62
		Fusion	0.63 / 0.63	0.63 / 0.62	0.64 / 0.63
10 s	*Dense ANN*	Contact	0.78 / 0.77	0.80 / 0.80	0.80 / 0.79
		Radar	0.75 / 0.75	0.76 / 0.76	0.75 / 0.74
		Fusion	0.79 / 0.79	0.80 / 0.78	0.79 / 0.78
	1D CNN	Contact	0.76 / 0.76	0.73 / 0.71	0.75 / 0.74
		Radar	0.74 / 0.74	0.72 / 0.71	0.74 / 0.73
		Fusion	0.79 / 0.78	0.75 / 0.75	0.78 / 0.78
20 s	*Dense ANN*	Contact	0.84 / 0.84	**0.88 / 0.87**	0.87 / 0.86
		Radar	0.81 / 0.81	**0.83 / 0.83**	0.82 / 0.81
		Fusion	0.86 / 0.84	0.87 / 0.87	**0.88 / 0.87**
	1D CNN	Contact	**0.85 / 0.85**	0.82 / 0.80	0.84 / 0.83
		Radar	**0.82 / 0.82**	0.80 / 0.78	0.81 / 0.80
		Fusion	**0.86 / 0.84**	0.82 / 0.82	0.85 / 0.85

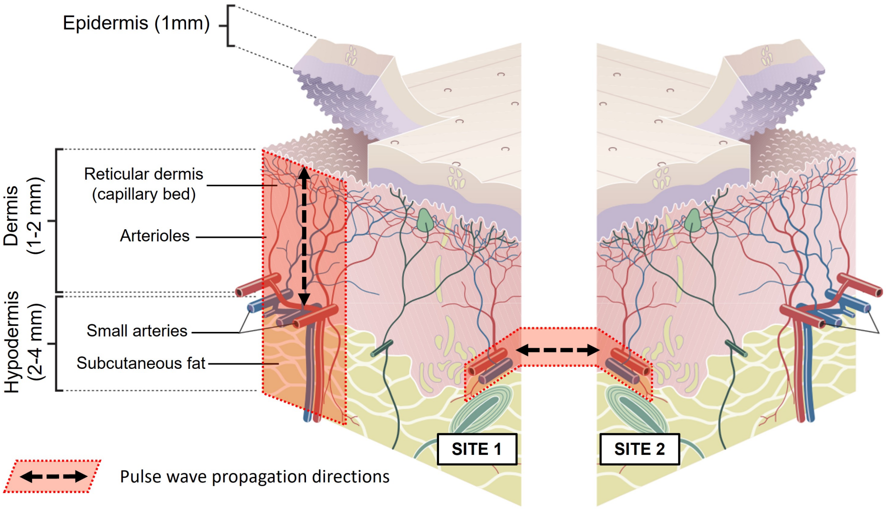


Figure 1: Layered structure of human skin, showing vascular presence and important structures. Pulse waves propagate through the vessels between different sites (horizontal arrow) and between different layers (vertical arrow).

allow for data-driven camera-independent channel separation, which in turn allows for precise measurement of MW PTTs [7]. Initially we checked the performance of existing blind source separation methods such as PCA and ICA. Our first proposed algorithm was based on camera physics – specifically quantum efficiency of the specific image sensor, using it to separate the bands. Subsequent variants were fully data driven, using the genetic algorithm (GA) paradigm to optimize parameters governing the linear combinations of channel mixtures, with different fitness functions – either phase delay (PD) between waveforms or error of a trained BP regressor [8]. These lead to good channel separation and MW PTT computation, enabling subsequent training of BP estimation models. Finally we confirmed on an in-house dataset with 13 subjects that such MW PTTs are well-correlated with BP, and trained RandomForest regression models to predict both systolic and diastolic BP in a leave-one-subject-out (LOSO) experiment with and without personalization. The best-performing models achieved errors within clinical standards, as shown in Table 2.

Table 2: Comparison of the MAEs in mmHg for SBP and DBP estimation when using different channel separation algorithms. We compare against the baseline of using no channel separation. We report results for experiments with and without personalization in a leave-one-subject-out (LOSO) experiment.

	General regressor [mmHg]			*Personalized regressor [mmHg]*		
Algorithm	**MAE$_{SBP}$**	**MAE$_{DBP}$**	**MAE$_{AVG}$**	**MAE$_{SBP}$**	**MAE$_{DBP}$**	**MAE$_{AVG}$**
Baseline	*11.31±1.50*	*9.02±1.60*	*10.17±1.55*	*8.64±1.62*	*6.12±1.48*	*7.38±1.55*
PCA	10.22±1.31	8.91±1.19	9.57±1.25	8.01±1.25	5.99±1.35	7.00±1.30
ICA	9.81±1.20	6.97±1.10	8.39±1.15	6.83±1.10	5.75±1.30	6.29±1.20
Physics (or.)	7.72±1.00	5.46±0.98	6.59±0.99	4.78±0.96	3.89±0.97	4.34±0.97
Physics (ref.)	6.94±1.02	5.03±0.96	6.06±1.01	4.00±0.94	2.88±0.99	3.39±1.00
GA-BP	**6.89±0.95**	**4.91±0.98**	**5.90±0.97**	**3.48±1.02**	**2.61±0.90**	**3.05±0.96**
GA-PD	7.02±1.03	4.97±0.97	6.00±1.00	4.01±0.98	3.03±0.96	3.52±0.97

4 Conclusion

Overall we showed that contact-free sensors leveraging the information from the EM spectrum are an affordable unobtrusive alternative to wearables, and can achieve similar performance in monitoring of important physiological parameters and states. While some limitations and challenges remain, such as difficult uncontrolled conditions and privacy concerns, there is potential for implementing the proposed methods in a single device, which could immensely improve the speed, cost and comfort of physiological monitoring both at home and in hospitals.

Acknowledgements

I would like to acknowledge the support of my supervisor prof. dr. Mitja Luštrek, my co-supervisor prof. dr. Wenjin Wang, Department of Intelligent Systems (E9), the Slovenian Research and Innovation Agency (ARIS), and Jožef Stefan International Postgraduate School.

References

[1] Matthew Smuck, Charles A Odonkor, Jonathan K Wilt, Nicolas Schmidt, and Michael A Swiernik. The emerging clinical role of wearables: factors for successful implementation in healthcare. *NPJ digital medicine*, 4(1):1–8, 2021. URL `https://doi.org/10.1038/s41746-021-00418-3`.

[2] He Liu, Yadong Wang, and Lei Wang. A review of non-contact, low-cost physiological information measurement based on photoplethysmographic imaging. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2088–2091. IEEE, 2012. URL `https://doi.org/10.1109/EMBC.2012.6346371`.

[3] Yongshen Zeng, Dongfang Yu, Xiaoyan Song, Qiqiong Wang, Liping Pan, Hongzhou Lu, and Wenjin Wang. Camera-based cardiorespiratory monitoring of preterm infants in nicu. *IEEE Transactions on Instrumentation and Measurement*, 2024. URL `https://doi.org/10.1109/TIM.2024.3395314`.

[4] Gašper Slapničar, Wenjin Wang, and Mitja Luštrek. Classification of hemodynamics scenarios from a public radar dataset using a deep learning approach. *Sensors*, 21(5):1836, 2021. URL `https://doi.org/10.3390/s21051836`.

[5] Sven Schellenberger, Kilin Shi, Tobias Steigleder, Anke Malessa, Fabian Michler, Laura Hameyer, Nina Neumann, Fabian Lurz, Robert Weigel, Christoph Ostgathe, et al. A dataset of clinically recorded radar vital signs with synchronised reference sensor signals. *Scientific data*, 7(1):291, 2020. URL `https://doi.org/10.1038/s41597-020-00629-5`.

[6] Gašper Slapničar, Wenjin Wang, and Mitja Luštrek. Feasibility of remote pulse transit time estimation using narrow-band multi-wavelength camera photoplethysmography. In *Adjunct Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2022 ACM International Symposium on Wearable Computers*, pages 115–117, 2022. URL `https://doi.org/10.1145/3544793.3560339`.

[7] Gašper Slapničar, Wenjin Wang, and Mitja Luštrek. Feasibility of remote blood pressure estimation via narrow-band multi-wavelength pulse transit time. *ACM Transactions on Sensor Networks*, 20(4):1–21, 2024. URL `https://doi.org/10.1145/3597302`.

[8] Gašper Slapničar, Wenjin Wang, and Mitja Luštrek. Generalized channel separation algorithms for accurate camera-based multi-wavelength ptt and bp estimation. *Biomedical Optics Express*, 15(5):3128–3146, 2024. URL `https://doi.org/10.1364/BOE.518562`.

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 communications 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 3 September 2024 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Informatica **4**8 (2024) Number 3, pp. 295–504