Optimizing Sequential Forward Selection on Classification Using Genetic Algorithm

Knitchepon Chotchantarakun
E-mail: knitchepon@go.buu.ac.th
Department of Information Studies, Faculty of Humanities and Social Science, Burapha University
169 Longhaad Bangsaen Rd, Saensuk, Mueang, Chonburi, 20131, Thailand

Keywords: classification accuracy, data mining, genetic algorithm, optimization, sequential feature selection

Received: June 22, 2023

Regarding the digital transformation of modern technologies, the amount of data increases significantly resulting in novel knowledge discovery techniques in Data Analytic and Data Mining. These data usually consist of noises or non-informative features that affect the analysis results. The features-eliminating approaches have been studied extensively in the past few decades name feature selection. It is a significant preprocessing step of the mining process, which selects only the informative features from the original feature set. These selected features improve the learning model efficiency. This study proposes a forward sequential feature selection method called Forward Selection with Genetic Algorithm (FS-GA). FS-GA consists of three major steps. First, it creates the preliminarily selected subsets. Second, it provides an improvement on the previous subsets. Third, it optimizes the selected subset using the genetic algorithm. Hence, it maximizes the classification accuracy during the feature addition. We performed experiments based on ten standard UCI datasets using three popular classification models including the Decision Tree, Naive Bayes, and K-Nearest Neighbour classifiers. The results are compared with the state-of-the-art methods. FS-GA has shown the best results against the other sequential forward selection methods for all the tested datasets with $O(n^2)$ time complexity.

Povzetek: Genetski algoritem je bil uporabljen za iskanje najboljšega zmanjšanega nabora atributov za namene strojnega učenja. FS-GA dosega boljše rezultate s tremi algoritmi na 10 UCI domenah.

1 Introduction

With the development in the field of Computer Science and Information Technology, data collection is a significant aspect that needs to be considered due to the hidden information behind these data. The amount of data is growing rapidly along with modern technologies both in dimension and volume. Usually, these data can be a high-dimensional dataset containing an excessively large number of variables or features. Some of these features may not contain valuable information and are regarded as noise. In the early stage of Data Mining (DM), which is the data preparation step, it is essential to eliminate these non-informative features from the large feature set. The removal of these features increases the learning performance and computational efficiency in the DM process. This dimensionality reduction technique is feature selection.

The process of feature selection is minimizing the redundant features and maximizing the relevant features by identifying a feature subset consisting of only informative features. It offers advantages such as reducing the requirement for computer storage, enhancing data visualization, improving model prediction, and reducing training times [1]. To guarantee the optimal feature subset, an exhaustive feature selection method requires $O(2^n)$ of time complexity, where n is the number of features in the input data. Exhaustive searches such as branch and bound [2, 3] result in exponential growth in computing time. Even though this method ensures the optimal feature subset, it is not feasible, particularly for a considerably large n. This NP-hard problem requires enormous execution time. Several search strategies have been studied for suboptimal solutions to reduce the time complexity. Feature selection algorithms are classified into different approaches. One of the standard approaches categorized it into three different types including filter method, wrapper method, and embedded method [4, 5].

The filter method concerns the relationship between the feature and the class label. It uses measurements such as similarity or distance to rank features from highest to lowest score. It concerns only the criterion value of individual features by neglecting the relationship between the feature and the classifier. Some examples of the filter methods are the Chi-square Test, Euclidean Distance, and Information Gain. The advantage of the filter method is the small computing time.

The wrapper method determines the goodness of each feature according to the classification accuracy, that is, the selected feature subset directly depending on the classification algorithm. The computing time is much slower than the filter method because of the application of the data mining algorithm on each considering subset

during the searching process. It is the main disadvantage of the wrapper method over the filter method. Some examples of wrapper methods are Sequential Feature Search and Heuristic Search. Regarding the quality of the selected feature subset, the wrapper method provides better performance since it is related to the classifiers used in classification.

The embedded or hybrid method combines feature selection with model training to reduce computing time. It returns the learned model and the feature subset concurrently. The embedded method applies a filter and a wrapper to select candidate feature subsets and then evaluates these subsets to find the best subset using classification. This method reduces the number of features in the dataset along with the decrease in computing time and leads to better performance.

Another categorization approach related to the training data before the learning process of feature selection is Supervised and Unsupervised algorithms [6]. In the supervised method, learning data are labeled with classes. On the other hand, the unlabeled data are applied to the unsupervised algorithm. The Unsupervised method arranges similar features into classes using some given criterion. The unsupervised method is usually more complicated than the supervised method.

The objective of this study is to explore an effective way to select a suboptimal feature subset by maximizing the selection performance in terms of classification accuracy regarding forward sequential feature selection. We discuss some related work in section 2. Section 3 presents the proposed method. Section 4 provides the experimental details of this study. Results and discussion are examined in section 5. The conclusion is stated in section 6.

2 Related work

2.1 Feature selection process

A very high dimension of collected data includes both relevant and irrelevant features in the dataset resulting in a vast number of feature selection techniques proposed in the literature. These dimensionality reduction techniques become significant by removing the irrelevant features from the relevant ones. Thus, feature selection is a necessary prior step for reducing the computation time and improving the DM performance. Feature selection consists of three essential steps name search, evaluate, and stop. It not only improves the classification accuracy by reducing irrelevant features but also reduces the computing time.

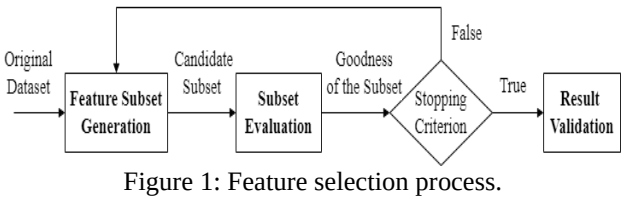

Figure 1: Feature selection process.

Figure 1 shows a feature selection process by first producing candidate feature subsets. Then, evaluate each candidate subset using the chosen strategies. This subset evaluation step can either be a filter or a wrapper approach. Continue the selection process until the stopping criterion is met. It can be a specific number of selected features, the number of predefined iterations, or there are no better solutions while adding a newly selected feature. Finally, compare the results using the chosen classifiers in the result validation step.

The search procedures can be categorized into three strategies [7]. The first strategy is the Exponential algorithm, in which the subset size grows exponentially. Some examples of this strategy are the Exhaustive search and the Branch-and-bound algorithm. The second strategy is a Sequential algorithm such as Sequential Forward Floating Selection (SFFS) [8]. This searching method adds or removes features from the current feature subset. The size of the subset can be increased or decreased depending on which direction produces a higher criterion value. The third strategy is the Random algorithm which commonly has a linear time complexity. It is designed to maximize the solutions and avoid the local optimum. The drawback of this strategy is the difficulty of choosing effective parameters. This Heuristic search optimizes the solution by incorporating randomness into the search process.

Evolutionary algorithms, as part of random algorithms, are becoming a more attractive field of study. Some remarkable examples include GA, Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). GA is developed from the Darwinian principle relating to the survival of living things. PSO is more efficient than GA but requires numerous mathematical calculations and user-defined parameters and is difficult when dealing with an experiment. ACO deals with the shortest paths discovered by the real ants while searching for their food. Even though the PSO and ACO algorithms produce similar results to GA, researchers take more interest in GA due to its simplicity and efficiency for implementation [9].

2.2 Sequential feature selection

The iterative nature of Sequential Feature Selection is to select a feature and add it to an active subset one by one sequentially. One of the earliest sequential selection methods is Sequential Forward Selection (SFS) [10]. It searches for the best feature from the remaining feature set in the forward direction. Hence, the selected subset gradually increases in size. The addition of a feature needs to maximize the performance validation by raising the learning accuracy of the active subset. Repeat the same operation until reaching the specified subset size. SFS is a base subset construction for other complicated selection algorithms.

The SFFS algorithm combines the forward SFS with the Sequential Backward Selection (SBS). The addition of SBS provides a better search than SFS by introducing the backtracking step in the backward direction. This backtracking step helps to eliminate a feature that affects learning efficiency. It is a conditional step where the improvement occurs during the selection process. SFFS is

widely used in several applications and is said to be the state-of-the-art (SOTA) method. The SFFS algorithm consists of three simple steps including the inclusion step (SFS step), the backtracking step (SBS step), and the stopping criterion checking step.

The improved version of SFFS occurs continuously starting from the Adaptive SFFS (ASFFS) [11]. ASFFS used SFFS as a baseline and improved the searching process by looking ahead both forward and backward for some specified number of features adaptively. It provided a higher opportunity to find a better subset. Additional work on the adaptive step significantly requires longer computing time.

The remarkable improvement of SFFS is the Improve Forward Floating Selection (IFFS) [12]. The idea of IFFS is to remove the weak feature in the selected subset and add a feature from the remaining set. Due to the fact that the best (k−1)-subset is not a fundamental subset for constructing the best k-subset. Hence, IFFS is able to fix the nesting problem. Not only the IFFS is capable of solving the nesting effect of SFFS but also provides a simpler algorithm than ASFFS. IFFS returns a better solution than SFFS and ASFFS with a slightly longer computing time than SFFS.

Similar to ASFFS, the Sequential Deep Floating Forward Search (SDFFS) [13] performed a deep search in both directions to prove the existence of better results than the current subset. This study compared SDFFS against SFS, SFFS, IFFS, and Plus-L-Minus-R (PlMr) [14]. The results showed that SDFFS returns the highest accuracy for the majority of the tested datasets. However, SDFFS required massive computing time with a small chance of finding a better subset.

Table 1: Summary Table for related methods on Sequential Feature Selection.

Study	Technique	Dataset	Performance Metric
[10]	**SFS**	Signal classification systems	Classification accuracy
[8]	**SFFS**	Not indicated	Mahalanobis distance
[11]	ASFFS	Mammogram and Sonar datasets	Bhattacharyya distance
[12]	IFFS	Mammogram, Sonar, Musk, and one versus six numeral recognition datasets	Classification accuracy, Mahalanobis distance, Divergence distance,
[13]	SDFFS	Colon, DLBCL, SRBCT, lung, musk1, synthetic, mfeat, and arrhythmia datasets	Classification accuracy
[15]	MLFI	Wine, Online shopper, Lymphography, Crowdsourced, Ionosphere, Soybean, Spectf heart, Sonar datasets	Classification accuracy
[16]	OFMB	Wine, Thoracic Surgery, Online shopper, Lymphography, Image Segmentation, Crowdsourced, Breast Cancer, Ionosphere, Soybean, Spambase, Sonar, Urban land cover datasets	Classification accuracy

Recently, two novel algorithms regarding sequential search were presented. Multi-levels Forward Inclusion (MLFI) [15] provided an improvement by applying an adaptive multi-level forward search method without any backtracking step. Whereas, the One-level Forward Multi-level Backward Selection (OFMB) [16] focused on the adaptive backtracking search direction. Most of the tested results from both algorithms showed higher accuracy than the previous standard methods. These two techniques are deterministic algorithms.

Table 1 summarizes some related popular sequential feature selection methods. The SOTA methods are indicated in bold. Other techniques are extensions of the SOTA method. The performance matrices applied to the proposed algorithm usually either be some kind of distance measure or classification accuracy. Many of them employ open datasets which are free and complete to test the goodness of their performances.

2.3 Genetic algorithm in feature selection

A deterministic approach for feature selection such as a floating search may be trapped in a suboptimal solution. Recently, many researchers have tried to explore a new area of feature selection techniques that are capable of escaping the suboptimal solutions using evolutionary techniques as part of the traditional method to improve performance. These hybrid approaches take advantage of the distinctive properties of at least two methods to optimize selection efficiency. The various hybrid methods optimize the solutions based on evolutionary computation related to biological evolution.

One of the most well-known evolutionary algorithms is GA [17]. The idea is to imitate the natural evolution process for survival based on the Darwinian principle for solving complicated computation problems. The GA process is motivated by natural evolution including inheritance, selection, crossover, and mutation. The GA concept has been implemented in various areas of computer science. One of them is the feature selection based on natural genetics that provides powerful search proficiency in large datasets.

GA is iteratively operated on a set of populations represented by chromosomes. In computational terms, these chromosomes represent binary strings which randomly generated via an encoding mechanism. In a binary string representation, a chromosome contains bit 1 or 0 which indicates the presence or absence of a particular feature in the active subset. The number of features in a subset implies the length of the chromosome. Each chromosome has a fitness value indicating its quality to rank them in descending order.

In the feature selection problem, the criterion function is assigned as a fitness function which is the classification accuracy. In each iteration, the fitness value is calculated for each candidate solution. Selection of higher fitness gives more chances of getting favorable solutions. The selected individuals represent the parents of the next generation. Crossover and mutation operators are applied to the parent chromosomes producing new populations in the search space. Since ‘good’ parents produce ‘good’

offspring, GA has a strong possibility that the solutions are expected to converge to the global optimal over time while the evolution process takes place for several generations.

Tournament Selection is one of the most common selection methods due to its simplicity and efficiency. It randomly selects a set of *k* individuals regarding their fitness. The highest *k* fitness is selected and assigned to the next generation. In crossover operation, two-parent chromosomes crossbred to construct two new chromosomes. One-point, two-point, or *k*-point crossovers are the random points for constructing children's chromosomes. The new chromosomes have a great opportunity to gain the dominant genes from the parent generating higher fitness. This crossover operation provides a good capability to solve the local optimum problem.

The other critical aspect of evolution is mutation since it also mitigates the risk of the search falling into a local optimum. This operation is applied to the evolutionary step in order to introduce new genes in the children's chromosomes. Mutation randomly changes the value of a gene by inverting the chosen pair of bits to the opposite values. A pair of flipping bits is essential to preserve the number of features in the active subset.

Several researchers integrate the GA technique with other methods when dealing with feature selection. In [18], the hybrid GA with PSO using random forest (RF) selected the best feature subset to predict the risk of heart disease. In [19], a floating search technique combined GA with SFFS to improve performance. It used Mutual Information (MI) as a criterion function to create a candidate subset during the inclusion step. In the GA step, it assigned a population size of 4-100 individuals, a mutation rate of 0.01, with a single-point crossover. The number of generations was 500. The experiments showed that GA improved the overall performance for most of the tested datasets. However, the inclusion of a backtracking step limited the search space.

In [20], to select a feature, they applied a machine learning (ML) model to construct a fraud detection engine on credit card data using GA. They tested the proposed method using various classifiers including Random Forest (RF), Decision Tree (DT), Naive Bayes (NB), Logistic Regression (LR), and Artificial Neural Network (ANN). The result showed that their proposed method exceeded the previous works.

The study from [21] presented the disease detection model using hybrid feature selection based on Honeybee-SMOTE and c4.5 algorithm relating to feature selection problems. Other research areas are the applications of feature selection techniques in education mining, text mining, and medical science [22-24].

2.4 Data mining in real-world applications

Data mining is an essential part of data analytics. Its techniques are used in many different fields such as business decision-making, bank insurance tasks, manufacturing, education, medicine, and public health. In the telecommunications business, results from data mining can improve customer services by focusing on finding patterns in customer behavior. In bank insurance, data mining can solve the management risk or customer exit problems. In manufacturing, data mining is used to predict phenomena that may occur in production and find solutions. In educational services, data mining algorithms are used to identify factors affecting student achievement and design policies to support them. In the medical and public health fields, data mining techniques are also used to solve various problems, such as predicting disease from patient symptoms or analyzing the risk factors for serious diseases. It can be seen that the use of data mining techniques has a widely used in large scope. Our proposed method is one of many feature selection techniques and is part of the data mining process, thus it can be applied to any real-world applications mentioned above.

3 Proposed method

This study is a wrapper-based approach that aims to optimize the sequential forward feature selection. The possible feature subset is generated using machine learning models to determine the goodness of each subset. The number of selected features in the active subset gradually increases or remains unchanged until the specified subset size is reached. In traditional floating search, the evaluation process appears in both forward and backward directions. The newly selected subset is a result of the previously selected subset. There is a possibility that the solution may be trapped in a local optimum. Moreover, the backtracking step also leads to a longer search time.

The proposed method tries to remove the weakness of the floating search algorithms by applying a randomized technique to avoid the local optimum and maximize the results by focusing only on the forward direction. A maximum *k*-subset is carefully selected from a large search space without decreasing the current feature size. Subsequently, a newly selected subset with a higher classification accuracy is explored.

We attempt to optimize a sequential forward selection that outperforms the standard methods. The combination of the feature forward selection, the weak feature replacement, and the GA technique come up with a novel forward feature selection method named Forward Selection with Genetic Algorithms (FS-GA). FS-GA intends to remove the backward step and explore further in the forward directions as well as escape the local optimal solution. FS-GA considers a larger search space that leads to a more exhaustive search. Hence, it increases the opportunity to optimize the active feature subset regarding the fitness value.

The mathematical calculation is as follows. Assume there is a feature set $Z = \{z_1, z_2, z_3, \ldots, z_D\}$, where D is the input features of the dataset. The subset $S_k = \{s_j \mid j = 1, 2, \ldots, k; s_j \in Z\}$ is the active subset, where $0 \leq k \leq D$ and d is the target subset size. Initialize $S_0 = \{\}$ and $k = 0$. Let s^+ be a selected feature included in the active subset since $s^+ = \arg\max F(s_k = s)$, where $s \in Z - S_k$ and F is the fitness value.

Algorithm 1: Forward Selection with Genetic Algorithms (FS-GA)

Input: A set of the input feature *Z*; *Fitness value F* represents a function to determine the quality of the chromosome; *d* is the target size; the *population* is a set of selected chromosomes.
Output: A feature subset max(S_k) with maximum classification accuracy.
Initialize: Initialize $S_0 = \{\}$; $k = 0$, *generation* = 1.
(1) Feature Inclusion Step
#Apply SFS to search for s^+ and max(S_k)
$s^+ = \arg\max F(s_k = s)$, where $s \in Z - S_k$
$S_{k+1} = S_k + s^+$, k +1
update max(S_k) = S_{k+1}
(2) Feature Improvement Step
#Improve the selected subset S_k iteratively.
repeat
for s_j in S_k : (j = 1 to k)
$S_{k-1} = S_k - s_i$
for s_i in $Z - S_{k-1}$: (i = 1 to $d - (k-1)$)
$s_i = \arg\max F(s_i)$
if $F(S_{k-1} + s_i) > F(S_k)$:
$S_k = S_{k-1} + s_i$
update max(S_k) = S_k
until $F(S_k)$ is the maximum feature subset
(3) Evolutionary Step
repeat
(3.1) Initialize Population
#Apply SFS to create a subset size 2*k*
length (*individual*) = 2*k*
$a = k$ +1
for a from k +1 to 2*k*:
$s^+ = \arg\max F(s_a = s)$, where $s \in Z - S_a$
$S_{a+1} = S_a + s^+$
$a = a + 1$
update max(S_a) = S_{a+1}
Randomly assign "1" to each gene in the chromosome for *k* genes; the other genes assign to "0" to get a binary string of 0 and 1 where both bits have equal *k* size. (This step applies only at the beginning of the process for the initial two chromosomes that represent the *population*)
(3.2) Selection Operation
#Apply the Rank Selection technique to select the best two chromosomes from the population according to the fitness value (*F*) of each candidate chromosome.
sort(*individual*(*F*))
*parent*1 = **select_parents**(*population*, *F*)
*parent*2 = **select_parents**(*population*, *F*)
(3.3) Crossover Operation
#Perform the One Point Crossover technique to randomly choose a point on the chromosome for swapping the genetic material
point = **random_point**(*individual*)
*child*1 = *parent*1[0: point] + parent2[point:]
*child*2 = *parent*2[0: point] + parent1[point:]
population = {*parent*1, *parent*2, *child*1, *child*2}
Control the number of bit 1 equal to *k*
(3.4) Mutation Operation
#Perform randomly swapping any bit of each chromosome by swap bit 0 with bit 1 or vice versa
for *individual* in *population*:
*index*1 = **random_index**(*individual*)
*index*2 = **random_index**(*individual*)
*index*1 and *index*2 must be different bit
temp = *individual*[*index*1]
individual[*index*1] = *individual*[*index*2]
individual[*index*2] = *temp*
generation = *generation* +1
until *generation* > 100
(4) Termination Condition
#Stop the execution when $k = d$
if $k < d$:
return the maximum subset max(S_k)
continue steps 1 to 4.
End

Algorithm 1 is a pseudocode of the FS-GA algorithm that illustrates the selection process for the candidate feature subset which is explained below.

Step 1: In the beginning, select a feature from the remaining feature set. Then add it to the active feature subset. Move on to step 2 with the feature subset S_k and increase the size of k by 1.

Step 2: Continue from the k-subset (S_k), and remove one feature to get k–1 subsets (S_{k-1}). Repeat this process k–1 times. Then, apply SFS to select a feature from the remaining set ($Z - S_{k-1}$). Add the newly selected feature to get S_k for k–1 subsets. Evaluate all candidate subsets to find an improvement. Replace the improved subset with the current subset then repeat step 2. Otherwise, continue step 3 with the maximum subset S_k.

Step 3: Apply GA to the selected subsets.

From the currently selected subset, we apply SFS to create a subset of size 2*k*. Then, randomly generates two individuals by assigning bit 1 to *k* genes and bit 0 to the other genes in the chromosomes. These two individuals or chromosomes represent the initial population as a starting point of the evolutionary step. This technique is used in only the first generation. Process the crossover and mutation to get four individuals consisting of two parents and two children chromosomes.

From the second generation onwards, we apply the selection operation by ranking the four individuals according to their finesses. Then, select only the best two individuals to be the parent chromosomes. If the two selected chromosomes have higher fitness values than the previous chromosome of size *k* with maximum fitness, then replace the previous chromosome with the newly selected one that is the best *k*-subset found so far.

At this point, we have two parents with the highest fitness values. Apply crossover operation to the selected parents to produce two children's chromosomes. Adjust the number of bit 1 in the offspring to preserve the subset size. Now, there are four individuals in the population pool.

Apply mutation operation to all individuals. Randomly select the first gene, then flip that gene to the opposite bit. Randomly select the second gene of the opposite bit to the first gene and flip it. Repeat step 3 for 100 generations, then go to step 4.

Step 4: Repeat steps 1 to 4 until $k = d$.

3.1 The application of genetic algorithm

The application of GA optimizes the search algorithm to discover the most desired solutions. Based on natural selection, construct the genes to form chromosomes. These chromosomes represent a population in the pool. GA basic operations include selection, crossover, and mutation.

Selection: The selection operator selects potentially useful solutions to recombine them using various techniques such as Tournament Selection, Rank Selection, or Random Selection. Tournament selection is applied in our study. All individuals in the population pool are measured using the fitness function. We rank them in descending order. The highest two fitness are selected for calculation in the next generation.

Crossover: Crossover operation is the matching of the two parent chromosomes producing the offspring chromosomes. This operation can be classified as a single-point, two-point, or uniform crossover. The crossover operator splits chromosome pairs randomly and combines the crossover pairs to form a pair of offspring chromosomes.

In our study, the one-point crossover is applied since it is the most well-known operation. The selected parent population is divided into two parts at a randomly selected point, namely the crossover point. Hence, the information from one gene to another is interchangeable. The genetic information from the two parents is exchanged to produce two children's chromosomes shown in Figure 2.

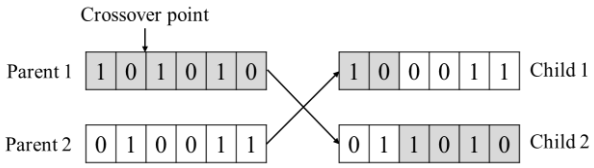

Figure 2: Crossover operation.

The idea behind the crossover operation is that the children's chromosomes have a better chance to carry out the good characteristics of their parents. In other words, the subsets of higher fitness are possibly discovered.

Mutation: Mutation operation introduces a slight variation in the chromosome by twisting one gene in the bit string. Practically, the mutation is done by randomly swapping any bit of the selected individuals. The mutation operator randomly selects genes and converts them to the opposite bit values on the binary string representation. This process changes the genetic sequence for introducing a new chromosome in the potential search space. It controls a cause of variation in the genetic population.

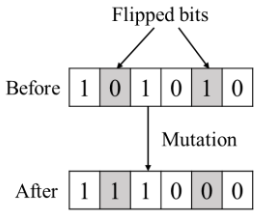

Figure 3: Mutation operation.

This study applies the interchanging mutation technique. Two positions are selected randomly from the bit strings and then switched the bit values of the genes. To stabilize the subset size, if the first randomly selected bit is 1, the second bit must be 0. Therefore, the size of the solution is preserved in every generation. Figure 3 illustrates the mutation operation.

3.2 Demonstration using the Wine dataset

The Wine dataset is one of the standard datasets from the UCI repository [25]. We have decided to use this dataset to show our calculation based on the FS-GA algorithm because of its simplicity and small. At the beginning, we have $Z = \{f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}\}$ with 13 features that means $d = 13$. We assign the generation to 100.

Feature Inclusion: Initially, assume that the algorithm applies forward selection for the first four features, that is $k = 1, 2, 3$, and 4. Then $S_1 = \{f_6\}$, $S_2 = \{f_6, f_{10}\}$, $S_3 = \{f_6, f_{10}, f_2\}$ and $S_4 = \{f_6, f_{10}, f_2, f_7\}$ respectively. At this point we have $k = 4$ is $S_4 = \{f_6, f_{10}, f_2, f_7\}$. This is the best 4-subset that we have found so far.

Feature Improvement: In this step, the subset size is $k = 4$, that is $S_4 = \{f_6, f_{10}, f_2, f_7\}$ with 89.98% accuracy. Remove a feature in the subset except for $s_4 = f_7$ to form three smaller subsets. Then the considering subsets are $\{f_6, f_{10}, f_7\}$, $\{f_6, f_2, f_7\}$, and $\{f_{10}, f_2, f_7\}$. Select the best feature in the remaining set ($Y - S_4$), then add it to the three candidate subsets above. At this point, there are three subsets of size 4 for comparison. After the F-value calculation, the algorithm finds a subset $\{f_{10}, f_2, f_7\}$ with the combination of a feature f_9 results in the highest fitness with 92.75%, that is $F(\{f_{10}, f_6, f_7, f_9\}) = 92.75\%$. Therefore, replace $\{f_6, f_{10}, f_2, f_7\}$ with $\{f_{10}, f_6, f_7, f_9\}$. Repeat this process with $S_4 = \{f_{10}, f_6, f_7, f_9\}$. The result is no better solution, thus move on to the evolutionary step with $S_4 = \{f_{10}, f_6, f_7, f_9\}$. The next step optimizes this solution using the GA technique.

Evolutionary Selection: Assume the algorithm continues until $k = 5$, that is $S_5 = \{f_{10}, f_6, f_7, f_9, f_5\}$ with 91.63% accuracy. Create a larger mating pool by selecting more features until the length of the feature subset doubles in size. Thus, the subset size must be $2k$, which is $2\times5 = 10$ features. Then, apply SFS to get a subset of size 10 which is $S_{10} = \{f_{10}, f_6, f_7, f_9, f_5, f_0, f_1, f_2, f_8, f_{11}\}$. Transform the feature subset into the binary strings where 0 and 1 indicate the absence and presence of the i^{th} feature in the solution. This step applies a random mechanism to create a wider solution region in the search space for the exploration by generating 2 parents in a mating pool. Individual representation using a binary string occurs only on the first iteration. From the second iteration onward, there would be 4 individuals, which are 2 parents and 2 children. These 4 individuals represent 4 chromosomes in the population set for the selection operation.

Parent 1 = $\{f_9, f_5, f_0, f_2, f_8\}$

f_{10}	f_6	f_7	f_9	f_5	f_0	f_1	f_2	f_8	f_{11}
0	0	0	1	1	1	0	1	1	0

Parent 2 = $\{f_{10}, f_9, f_5, f_0, f_2\}$

f_{10}	f_6	f_7	f_9	f_5	f_0	f_1	f_2	f_8	f_{11}
1	0	0	1	1	1	0	1	0	0

Figure 4: Two-parent selection.

1) Selection Operation: For the selection operation, there are 4 individuals in the mating pool in every iteration. The best 2 individuals are chosen according to the fitness values calculated by the selected classifier. Assume that the best 2 individuals are [0, 0, 0, 1, 1, 1, 0, 1, 1, 0] and [1, 0, 0, 1, 1, 1, 0, 1, 0, 0]. Figure 4 shows the representation of the parent sets {f_9, f_5, f_0, f_2, f_8} and {f_{10}, f_9, f_5, f_0, f_2} for the crossover operation.

2) Crossover Operation: In this step, a pair of parent chromosomes {f_9, f_5, f_0, f_2, f_8} = [0, 0, 0, 1, 1, 1, 0, 1, 1, 0] and {f_{10}, f_9, f_5, f_0, f_2} = [1, 0, 0, 1, 1, 1, 0, 1, 0, 0] perform crossover operation to produce a pair of child chromosomes. First, compute a one-point crossover by randomly choosing a crossover point from the parent. This point is located between two consecutive bits and then partitions each chromosome into two sections. Assume we have a crossover point of 4. After that, swapping the opposite parts of the two chromosomes forms the other two new chromosomes shown in Figure 5. Children's chromosomes may have more or fewer bits than the parents. In this case, random bits automatically flip to maintain the size of their parents which is a subset size 5.

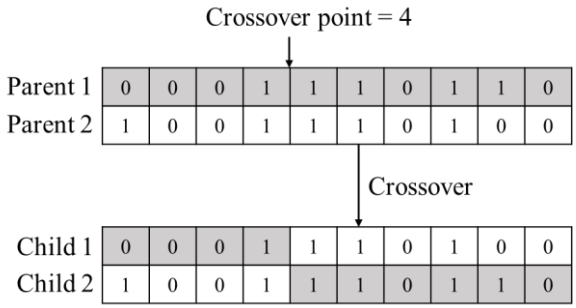

Figure 5: Perform one-point crossover.

3) Mutation Operation: The mutation operation applies to the child's chromosomes from the previous step. It randomly flips bits in the new chromosomes by turning 1 to 0 and vice versa.

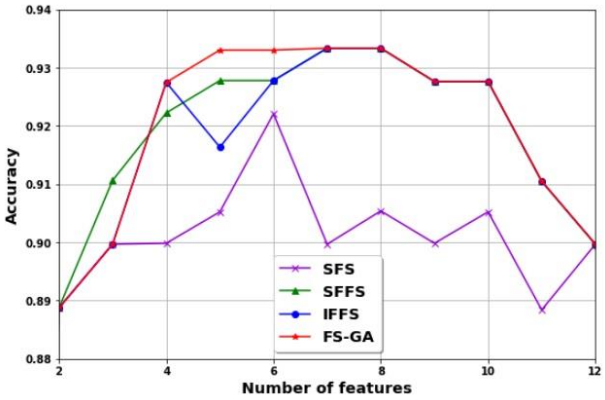

Figure 6: Classification accuracy of the Wine dataset.

The evolutionary step continues for several iterations that indicate the number of generations. The best fitness from the reproduction subsets is the one that survives for selecting the next feature. Figure 6 shows that S_5 from the FS-GA algorithm produces the highest classification accuracy with 93.3%, whereas the other sequential searching techniques cannot find this accuracy value. To explore even further, we can increase the number of generations and more features for gene representation results in a thorough search. Our study explores the potential subsets for 100 generations and then assigns the maximum solution of S_5. Return to step 1 and repeat the process for k = 6.

Termination Condition: The proposed algorithm processes sequentially by adding one feature on each iteration. The size is growing from k to d with the best performance for each subset size indicated by S_k. The algorithm stops when we get the required subset size. This method applies the concept of the evolutionary searching technique named GA to optimize the solutions in terms of classification accuracy using various machine learning models.

The FS-GA algorithm gives a higher chance to explore deeper to find the potential subsets that have not come across before and cannot be found by the earlier methods. There is a high possibility that better subsets with better accuracy are discovered. Therefore, the results move closer to the optimal solutions. The proposed algorithm is the sequential forward selection using the evolutionary method to improve performance. Due to the non-heuristic behavior, it is capable of avoiding the trap of the local optimum and providing suboptimal solutions that overcome the SOTA methods.

4 Experimental setup

The proposed algorithm was developed using Python programming language with the Jupyter Notebook editor to perform the experiments. We used ten standard datasets from the UCI listed in Table 2. These datasets are open data with adequate information for carrying out the experiments, and some of them were also used in previous works regarding sequential feature selection. Other kinds of datasets with CSV format are also applicable to the proposed algorithm.

Table 2: Tested Datasets.

Dataset Name	Instances	Dimensions
Wine	173	13
Thoracic Surgery	470	17
Lymphography	142	18
Image Segmentation	2310	19
Breast Cancer	569	32
Ionosphere	351	34
Soybean	307	35
Spam base	4601	57
Sonar	208	60
Urban Land Cover	675	147

We compared our results with SFS, SFFS, and IFFS using DT, NB, and K-Nearest Neighbor (KNN) for performance validation. These popular classifiers are implemented in many applications due to their dominant characteristics including robustness, effectiveness, and ease of implementation. FS-GA is a general selection method similar to the other previous methods which

means it can be applied to other kinds of datasets that contain structured information on both features and class. We applied the three classifiers to all four methods for the comparison. The evaluation was based on data normalization using 5-fold cross-validation. Noise data like missing values or outliers were eliminated where necessary.

5 Results and discussion

5.1 Compare FS-GA with the standard methods using DT

In Table 3, the experimental results of our proposed method are compared to the other popular sequential search methods. The comparison of maximum accuracy in percentage (%) and the number of selected features is in the brackets. The highest accuracy is highlighted in bold. The best solutions are the result of the FS-GA algorithm based on the DT classifier. SFS and SFFS cannot provide the maximum classification accuracy in all tested datasets. IFFS generates feature subsets identical to the proposed method in only Wine and Lymphography datasets. While the dimension increases, the FS-GA performance is the best among the other popular methods on the remaining datasets excluding only Wine and Lymphography sets. The experiments have proved that our proposed method is a more effective method than the other standard methods for forward feature selection due to the application of weak feature replacement and genetic optimization.

Table 3: Classification Accuracy using DT.

Dataset Name	SFS	SFFS	IFFS	FS-GA
Wine (13)	92.76 (3)	93.32 (6)	**93.89 (4)**	**93.89 (4)**
Thoracic Surgery (17)	80.64 (4)	81.28 (6)	81.06 (6)	**81.70 (6)**
Lymphography (18)	85.94 (4)	86.63 (5)	**86.65 (6)**	**86.65 (6)**
Image Segmentation (19)	84.29 (12)	85.24 (9)	87.14 (8)	**87.62 (10)**
Breast Cancer (32)	95.60 (5)	95.60 (5)	96.31 (7)	**96.49 (10)**
Ionosphere (34)	91.75 (20)	92.03 (15)	93.45 (7)	**94.02 (20)**
Soybean (35)	89.84 (20)	90.60 (13)	90.98 (15)	**92.11 (20)**
Spam base (57)	90.33 (17)	90.87 (16)	91.00 (19)	**91.07 (18)**
Sonar (60)	77.05 (6)	81.35 (10)	87.51 (10)	**87.99 (9)**
Urban Land Cover (147)	79.41 (20)	77.63 (17)	80.15 (13)	**81.19 (19)**

5.2 Compare FS-GA with the standard methods using NB

In Table 4, the classification accuracy enlarges by the proposed method based on the NB classifier. Only the Wine dataset has the same results for all comparison methods. Excluding the Wine dataset, IFFS produces equal maximum accuracy as the proposed method for Image Segmentation, Breast Cancer, Ionosphere, and Soybean datasets. In the Ionosphere dataset, although IFFS has the same accuracy as the FS-GA algorithm, it requires a higher number of features (14 > 10 features) consequently the proposed method considers to be the best performance for this particular dataset. In the remaining tested datasets, the proposed method achieves better results. Accordingly, FS-GA generates the minimal feature subset with the highest classification accuracy in the most tested datasets.

Table 4: Classification Accuracy using NB.

Dataset Name	SFS	SFFS	IFFS	FS-GA
Wine (13)	**93.30 (5)**	**93.30 (5)**	**93.30 (5)**	**93.30 (5)**
Thoracic Surgery (17)	85.11 (1)	85.11 (1)	85.11 (1)	**85.32 (5)**
Lymphography (18)	86.58 (8)	87.24 (10)	87.32 (9)	**88.03 (8)**
Image Segmentation (19)	81.90 (5)	81.90 (5)	**82.86 (5)**	**82.86 (5)**
Breast Cancer (32)	95.43 (8)	95.43 (8)	**96.14 (6)**	**96.14 (6)**
Ionosphere (34)	92.58 (14)	93.44 (11)	93.72 (14)	**93.72 (10)**
Soybean (35)	83.86 (20)	84.61 (15)	**91.73 (12)**	**91.73 (12)**
Spam base (57)	72.55 (4)	74.33 (13)	76.83 (11)	**81.03 (16)**
Sonar (60)	78.85 (19)	82.20 (17)	81.32 (9)	**82.71 (12)**
Urban Land Cover (147)	72.44 (15)	74.22 (14)	75.70 (14)	**76.00 (19)**

5.3 Compare FS-GA with the standard methods using KNN

As shown in Table 5, it is obvious that the classification accuracy is enhanced by the GA step based on the KNN classifier. FS-GA shows the maximum performance in almost all the tested datasets. In the Wine dataset, FS-GA produces equal solutions with SFFS and IFFS because the dataset is small with limited subset combinations. In the Thoracic Surgery dataset, even though IFFS has the same maximum accuracy as SFFS and FS-GA, it has a higher number of selected features, thus SFFS and FS-GA perform better in this case. In the Lymphography datasets, SFFS and FS-GA are the best methods. In Image Segmentation and Ionosphere datasets, IFFS performs quite well and has equal maximum accuracy with our method. For the rest of the results, while the feature size increases, the FS-GA algorithm produces the highest performance among the other techniques.

Table 5: Classification Accuracy using KNN.

Dataset Name	SFS	SFFS	IFFS	FS-GA
Wine (13)	92.21 (6)	**93.33 (7)**	**93.33 (7)**	**93.33 (7)**
Thoracic Surgery (17)	84.89 (5)	**85.96 (9)**	85.96 (10)	**85.96 (9)**
Lymphography (18)	86.55 (3)	**90.12 (11)**	88.72 (12)	**90.12 (11)**
Image Segmentation (19)	80.95 (10)	80.95 (7)	**81.43 (8)**	**81.43 (8)**
Breast Cancer (32)	88.09 (20)	88.66 (20)	88.80 (20)	**88.97 (20)**
Ionosphere (34)	95.43 (18)	95.43 (8)	**95.43 (6)**	**95.43 (6)**
Soybean (35)	93.45 (5)	94.02 (12)	94.59 (12)	**94.87 (9)**
Spam base (57)	89.10 (18)	88.73 (16)	90.99 (20)	**91.35 (20)**
Sonar (60)	88.92 (20)	91.26 (18)	92.09 (19)	**92.13 (20)**
Urban Land Cover (147)	79.36 (12)	79.84 (11)	79.34 (17)	**84.08 (17)**

5.4 Compare FS-GA using different classifiers

In Table 6, the comparison of FS-GA is presented by applying different criterion functions using DT, NB, and KNN. DT gives the highest accuracy on only three tested datasets related to Wine, Image Segmentation, and Breast Cancer datasets. Most of the highest accuracies are from the KNN classifier on the seven remaining sample datasets. On the other hand, using the NB classifier does not provide the best results in most cases. Accordingly, different classifiers affect the algorithm's performance. They return different results since each has a unique characteristic that allows access to the feature subset selection. Moreover, KNN returns the highest accuracy compared with the other two classifiers. Hence, KNN is a desired classifier for capturing the best solutions.

Table 6: Classification Accuracy of FS-GA using DT, NB, and KNN.

Dataset name	DT	NB	KNN
Wine (13)	**93.89 (4)**	93.30 (5)	93.33 (7)
Thoracic Surgery (17)	81.70 (6)	85.32 (5)	**85.96 (9)**
Lymphography (18)	86.65 (6)	88.03 (8)	**90.12 (11)**
Image Segmentation (19)	**87.62 (10)**	82.86 (5)	81.43 (8)
Breast Cancer (32)	**96.49 (10)**	96.14 (6)	88.97 (20)
Ionosphere (34)	94.02 (20)	93.72 (10)	**95.43 (6)**
Soybean (35)	92.11 (20)	91.73 (12)	**94.87 (9)**
Spam base (57)	91.07 (18)	81.03 (16)	**91.35 (20)**
Sonar (60)	87.99 (9)	82.71 (12)	**92.13 (20)**
Urban Land Cover (147)	81.19 (19)	76.00 (19)	**84.08 (17)**

The FS-GA algorithm provides an evolutionary optimization result to a small effective feature subset. The improvement is reinforced with the application of the GA due to a more in-depth search with more probability of discovering the optimal solution. Similar to other methods, FS-GA can also perform on larger datasets with longer computing time. The proposed algorithm identifies a more relevant and informative feature from the initial dataset by incorporating the feature improvement step with the evolutionary step using GA. The significant characteristic of GA as a randomized-based algorithm is to avoid the situation where the solutions are trapped in the local optimum. Moreover, there is no pattern learned from GA, that is each execution may obtain different results. Hence, it may need to apply the algorithm a few times to get the best performance.

5.5 Time complexity

The time complexity for the FS-GA algorithm derives from two critical steps. The first one is the feature improvement step which removes a feature from the active subset and adds another one to the same set. There are n features from the remaining set to be selected at most. This same operation repeats for n iterations. Thus, the feature improvement step requires at most n^2.

The second one is the evolutionary step. The computing time is directly related to the number of generations. In this situation, the number of generations that are assigned to FS-GA is 100, consequently, the computing time is $100*k$, where k is the size of the chromosome. The number of generations can be varied depending on our decision.

Apart from the two steps mentioned earlier, other actions are constant time and can be ignored. Therefore, the time combination of the two critical steps is $O(n^2 + 100*k)$. Consequently, FS-GA requires a little higher computation time than IFFS, since IFFS requires at most $O(n^2)$ time due to the weak feature replacement step. SFS is simply adding one feature at a time and leads to only $O(n)$. Moreover, SFFS is adding or removing one feature while increasing in size up to n feature with backward tracking at the most k times. Hence, SFFS is bounded by $O(kn)$ which is roughly $O(n)$ time complexity.

6 Conclusion

This study proposed a novel sequential forward selection approach based on GA. The proposed algorithm is Forward Selection with Genetic Algorithms (FS-GA). The aim is to maximize the classification accuracy and outperform various standard methods. The proposed method is a searching technique in the forward direction by improving the performance of SFS.

FS-GA incorporates a feature improvement step with GA to find the optimal subsets. The algorithm utilizes an evolutionary technique to optimize the solutions by adjusting the number of reproductions and the population size. GA improves the results not only by maximizing the classification accuracy but also by helping the searching process escape from the local optimum trapping. Therefore, with this powerful technique, the proposed method can search through possible feature subsets more exhaustively. Consequently, there is a greater chance of discovering a better subset from the candidate subsets.

We applied the DT, NB, and KNN classifiers to our experiments, and then compared the proposed method with the standard SFS, SFFS, and IFFS methods. The results showed that FS-GA performed the best for all the sample datasets with $O(n^2)$ time complexity.

The limitation of FS-GA is similar to the other evolutionary algorithms due to their random nature and does not guarantee the best results for the newly discovered subset. However, we can adjust the parameters such as the number of genes or generations to improve the results. Future directions can focus on modifying the number of generations or populations and altering the size of the individual.

References

[1] Zeng, Z., Zhang, H, Zhang, R. and Zhang, Y. (2014). Hybrid Feature Selection Method based on Rough Conditional Mutual Information and Naïve Bayesian

Classifier. *Hindawi Publishing Corporation*, ISRN Applied Mathematics. https://doi.org/10.1155/2014/382738

[2] Somol, P., Pudil, P. and Kittler, J. (2004). Fast Branch & Bound Algorithms for Optimal Feature Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7), pp. 900-912. https://doi.org/10.1109/tpami.2004.28

[3] Nakariyakul, S. and Casasent, D. P. (2007). Adaptive branch and bound algorithm for selecting optimal features. *Pattern Recognition Letters*, 28, pp. 1415-1427. https://doi.org/10.1016/j.patrec.2007.02.015

[4] Cai, J., Luo, J., Wang, S. and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, pp. 70-79. https://doi.org/10.1016/j.neucom.2017.11.077

[5] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, 40, pp. 16-28.

[6] Sutha, K. and Tamilselvi, J. J. (2015). A Review of Feature Selection Algorithms for Data Mining Techniques. *International Journal on Computer Science and Engineering (IJCSE)*, pp. 63-67.

[7] Jovic, A., Brkic, K. and Bogunovic, N. (2015). A review of feature selection methods with applications. *International Convention on Information and Communication Technology*.

[8] Pudil, P., Novovicova, J. and Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, pp. 1119-1125. https://doi.org/10.1016/0167-8655(94)90127-9

[9] Pavya, K. and Srinivasan, B. (2017). Feature Selection Techniques in Data Mining: A Study. *International Journal of Scientific Development and Research (IJSDR)*, 2(6), pp. 594-598.

[10] A. W. Whitney. (1971). A Direct Method of Nonparametric Measurement Selection. *IEEE Transactions on Computers*, pp. 1100-1103. https://doi.org/10.1109/t-c.1971.223410

[11] Somol, P., Pudil, P., Novovicova, J. and Paclik P. (1999). Adaptive floating search methods in feature selection. *Pattern Recognition Letters*, pp. 1157-1163. https://doi.org/10.1016/s0167-8655(99)00083-5

[12] Nakariyakul, S. and Casasent, D. P. (2009). An improvement on floating search algorithms for feature subset selection. *Pattern Recognition*, pp. 1932-1940. https://doi.org/10.1016/j.patcog.2008.11.018

[13] Lv, J., Peng, Q. and Sun, Z. (2015). A modified sequential deep floating search algorithm for feature selection. *International Conference on Information and Automation*, pp. 2988-2933. https://doi.org/10.1109/icinfa.2015.7279800

[14] Pudil, P., Ferri, F. J., Novovicova, J. and Kittler, J. (1994). Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions. *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pp. 279-283. https://doi.org/10.1109/icpr.1994.576920

[15] Chotchantarakun, K. and Sornil, O. (2021). An Adaptive Multi-levels Sequential Feature Selection. *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, 13, pp. 010-019.

[16] Chotchantarakun, K. and Sornil, O. (2021). Adaptive Multi-level Backward Tracking for Sequential Feature Selection. *Journal of ICT Research and Applications*, 15, pp. 1-20. https://doi.org/10.5614/itbj.ict.res.appl.2021.15.1.1

[17] Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press: Cambridge, UK. https://doi.org/10.7551/mitpress/1090.003.0016

[18] El-Shafiey, M. G., Hagag, A., El-Dahshan, E. A. and Ismail, M. A. (2022). A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest. *Multimedia Tools and Applications*, 81, pp. 18155-18179. https://doi.org/10.1007/s11042-022-12425-x

[19] Homsapaya, K. and Sornil, O. (2017). Improving Floating Search Feature Selection using Genetic Algorithm. *Journal of ICT Research and Applications*, 11(3), pp. 299-317. https://doi.org/10.5614/itbj.ict.res.appl.2017.11.3.6

[20] Ileberi, E., Sun, Y. and Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data*, 9(24). https://doi.org/10.1186/s40537-022-00573-8

[21] Aswal, S., Jyothi, A. and Mehra, R. (2023). Feature Selection Method Based on Honeybee-SMOTE for Medical Data Classification. *Informatica*, 46(9), pp. 111-118. https://doi.org/10.31449/inf.v46i9.4098

[22] Alija, S., Beqiri, E., Gaafar, A. S. and Hamoud, A. K. (2023). Predicting Students Performance Using Supervised Machine Learning Based on Imbalanced Dataset and Wrapper Feature Selection. *Informatica*, 47(1), pp. 11-20. https://doi.org/10.31449/inf.v47i1.4519

[23] Al-jadir, I., Wong, K. W., Fung, C. C. and Xie, H. (2017). Text Document Clustering Using Memetic Feature Selection. *Proceedings of the 9th International Conference on Machine Learning and Computing (ICMLC)*, pp. 415-420. https://doi.org/10.1145/3055635.3056603

[24] Panda, D., Panda, D., Dash, S. R. and Parida, S. (2021). Extreme Learning Machines with Feature Selection Using GA for Effective Prediction of Fetal Heart Disease: A Novel Approach. *Informatica*, 45(3), pp. 381-392. https://doi.org/10.31449/inf.v45i3.3223

[25] Dua, C. Graff. (2019). UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science. https://archive.ics.uci.edu/datasets