Adv. Math. Fin. App., 2026, **11**(1), P. 35-60

**Advances in Mathematical Finance & Applications**
**www.amfa.iau-arak.ac.ir**
**Print ISSN:** 2538-5569
**Online ISSN:** 2645-4610
**Doi:**10.71716/amfa.2026.01208255

Case Study

# Finding all Redacts in Financial Information Systems Based on Neighbourhood Rough Set Theory for Finance Data with Decision Makers Point of View

Soodabeh Amin[a], Seyed Majid Alavi[a,*], Parvaneh Mansouri[a], Abolfazl Saeidiafar[a]

[a] *Department of Mathematics and Computer Science, Ar.C., Islamic Azad University, Arak, Iran*

Article Info
Article history:
Received 2025-05-30
Accepted 2025-08-14

Keywords:
Neighborhood Rough Set Theory(NRST)
Feature Selection
Financial Information Systems(FIS)

ABSTRACT

The Neighborhood Rough Set (NRST) method is a valuable approach for selecting a subset of features from a complete dataset, enabling us to preserve the essential information that the entire feature set provides. In financial datasets, which often contain high-dimensional input features, effective feature selection techniques are crucial to identify the features that yield the most predictable results. In this work, we use neighborhood concepts to discover data dependencies and reduce the number of features in a financial dataset based solely on the data itself, without relying on additional information. This process also includes removing extra features. To facilitate a simple algorithm, we use the properties of neighbourhood rough sets to formulate a Binary Integer Linear Programming (BILP) model. Optimal solutions to these problems are obtained using genetic algorithms. Our approach allows for feature reduction from minimum to maximum cardinality. We demonstrate the efficiency of our proposed method compared to other techniques through various tables showing the results on several benchmark datasets characterized by unbalanced class distributions. The financial dataset used in the present study is taken from the UCI Machine Learning Repository.

## 1 Introduction

Nowadays, a broad amount of data is created every day and analyzing the data is a serious challenge. By data mining, we refer to techniques in different fields such as information technology, mathematical science, and statistical analysis [30,40]. Data mining technique are useful to analyze, comprehend, and

* Corresponding author. Tel.: +989183631557
E-mail address: *sm.alavi@iau.ac.ir*

visualize large sums of data kept at data warehouses, databases, or other types of data repositories [31]. Through data mining techniques, we can handle extensive datasets including countless number of features. The big number of features that such datasets represent creates issues for data miners as some of them might be irrelevant to the data mining techniques. These irrelevant features can degrade the quality of the results gained through data mining, which in turn lowers the opportunity to find useful knowledge out of the dataset. Feature selection is one of the approaches to handle such features [18]. The big data are utilized for accurate classification in science and engineering fields like astronomy, medicine and so on. Still, such datasets feature insignificant redundant and nosy characteristics. These features may decrease the classifier efficiency. Selecting the proper features is important to solve the issue. Thus, in many fields of study, the choice of feature has a key role to play [12]. Selection of features can be a way to lower the redundant and irrelevant features and have a higher clustering accuracy and performance [11]. In real-world applications, the distribution of classes in data collected from road traffic, medicine, credit cards, banking transactions, and the stock market may be uneven. Clearly, it is not possible to discover the hidden knowledge out of real-world datasets before developing novel and optimal techniques for feature selection in imbalanced data. Imbalanced data affects the performance of predictive models [6]. So far, various algorithms have been introduced for the selection of features in imbalanced data. In many studies, the search power of meta-heuristic algorithms has been used in algorithms designed to select features [28,41]. Particle Swarm Optimization (PSO) [23], Differential Evolution (DE) [24], Gravitational Search Algorithm (GSA) [40], Water Cycle Algorithm (WCA) [15], Forest Optimization Algorithm (FOA) [16], Cuckoo Search (CS) [33], Lightning Search Algorithm (LSA) [43], Monarch Butterfly Optimization(MBO) [44] and etc. are some of the meta-heuristics employed to solve function choice problem [41,42]. There had been a success instances of the usage of meta-heuristic algorithms for specific engineering and medical troubles like optimized connection weights with inside the neural network [5], Numerical Optimization [25], cloud computing [1], and stock market index prediction [20]. Due to the advancement of soft computing, researchers in finance, computer science, and mathematics have paid great attention to optimization research [6]. In addition, several algorithms for selecting features have been proposed to diagnose, classify, categorize, and detect patterns are available [2,3,3]. Following the introduction of rough set theory in 1981 by Pawlak [35], it has been used as a successful technique to select features in classified data. It is used in artificial intelligence and cognitive sciences in several different fields including knowledge discovery, expert systems, inductive reasoning, decision-making, intelligent systems, data mining, information systems, pattern recognition, machine learning, process control, and so on [46]. Using rough set theory, we can handle uncertainty, vagueness, and imprecision, a technique that is widely used for dimension reduction [45]. A key use of rough set theory is attribution reduction, which means eliminating redundant attributes without losing information [29]. In recent years, various researchers have used rough set theory combined with other methods to reduce and select features [26]. Recently, many studies have used rough set theory and fuzzy neighborhood theory combined with other methods in feature selection [4,9,14,21,22]. The paper is structured as follows: Section 2 includes the fundamental concepts of the rough set and the neighborhood rough theory. Section 3 details the proposed method. Sections 4 and 5 present the simulation results and analyze the experimental findings, respectively. Finally, the concluding section summarizes the key insights and implications of the study.

## 2 Preliminaries

Feature selection preserves essential information by reducing the dimensionality of financial data.

Rough set theory is a soft computing tool with various applications in data science. Data mining is one of the areas where rough set theory is used. There are several studies that prove that rough set theory is a popular and practical tool in feature selection. Most traditional RST-based feature selection methods depend on reduction. In the following, we review the research conducted in the field of feature selection using neighbourhood rough and rough set theory.

## 2.1. Rough Set Theory

Rough Set Theory (RST) is a mathematical framework for dealing with uncertainty and vagueness in data analysis. It was introduced by Zdzisław Pawlak [36] in the early 1980s and has since become a significant approach in various fields, including data mining, machine learning, and decision-making. RST provides tools for handling imprecise or incomplete information without requiring prior probability distributions.

## 2.2. Information and decision systems

An information system is a pair $S = (U, A)$, where : $U$ is a non-empty finite set of objects (the universe). $A$ is a set of attributes (features), where each attribute $a \in A$ is associated with a function $f_a: U \to V_a$ , mapping objects to their attribute values. $V_a$ is the value set of attribute $a$.

## 2.3. Indiscernibility

For any subset of attributes $B \subseteq A$ , the indiscernibility relation $IND(B)$ groups objects that have the same values for the attributes in $B$ :[36]:

$$IND(B) = \{(x, x') \in U^2 : \forall a \in B, a(x) = a(x')\}. \quad (1)$$

If $(x, x') \in INDS(B)$, then $x$ and $x'$ are indiscernible by attributes in $B$. The equivalence classes of the $B - indiscernibility$ relation are given $[x]_B$.

## 2.4. Lower and Upper Approximations

Let $S = (U, A), X \subseteq U$ and $B \subseteq A$. $X$ be estimated based on the information within $B$ by constructing the B-lower and B-upper approximations of X:

$$\underline{B}X = \{x \in U \mid IND(B)(x) \subseteq X\}. \quad (2)$$

$$\overline{B}X = \{x \in U \mid IND(B)(x) \cap X \neq \emptyset\}. \quad (3)$$

As mentioned above, the upper approximation set has all elements needed to classify it as $U$. The lower approximation has the minimum set of the feasible elements of $U$. It is such a tuple $\langle \underline{BX}, \overline{BX} \rangle$ that is named a rough set [34].

### 2.4.1. Feature Dependency and Significance

In Rough Set Theory, the concept of positive region is crucial for understanding how certain attributes can classify objects into specific classes. The positive region, denoted as $POS_C(D)$ , refers to the set of objects that can be definitively classified into a target set based on the information provided by a given set of attributes. The positive region with respect to the decision attribute $C$ and the subset $D$ is defined as:

$$POS_C(D) = \{ x \in U \mid IND(B)(x) \subseteq D \} \quad (4)$$

$IND(B)$ represents the indiscernibility relation defined by a subset of attributes $B$ (which could include all attributes or a selected subset). The set $POS_C(D)$ consists of those objects in the universe $U$ whose equivalence class (under the indiscernibility relation) is entirely contained within the target set $D$. The quality of classification in Rough Set Theory is another key concept that evaluates the accuracy and predictive capability of classification models. Rough Set Theory is recognized as an analytical tool for processing data and extracting information from it without requiring prior assumptions about the distribution of the data.

$$k = \gamma_C(D) = \frac{|POS_C(D)|}{|U|} \tag{5}$$

Where the numerator $|POS_C(D)|$ counts the number of objects that can be positively classified into the target set $D$. $|U|$ contains the total number of objects in the universe. The value of $k$ varies between 0 and 1.

1. If $k = 1$ : which means that all items in the universe may be definitely categorized into the target set D, the classification is complete.
2. If $k = 0$ : This indicates that none of the objects in the universe can be classified into $D$ .
3. A higher value of $k$ indicates better classification quality, meaning a greater proportion of objects can be successfully classified into the desired class.

The significance of an attribute can be evaluated primarily based on its ability to help distinguish between different classes. An attribute is considered significant if its presence improves the classification quality. The importance of an attribute $a$ relative to a decision attribute $C$ can be quantitatively assessed using various measures, such as:

$$\sigma_C(D, a) = \gamma_C(D) - \gamma_{(C-\{a\})}(D). \tag{6}$$

Where $\gamma_C(D)$ is the classification accuracy when attribute $a$ is included and $\gamma_{(C-\{a\})}(D)$ is the classification accuracy when attribute $a$ is excluded.

### 2.4.2. Reducts and Core

A reduct of a set of features (attributes) in a dataset is a minimal subset of features that preserves the classification ability of the original set. In other words, if $A$ is the original set of attributes and $R$ is a reduct, then: $R \subseteq A$ and the classification using $R$ is equivalent to that using $A$.The core is the intersection of all reducts of a set of features. It consists of those attributes that are essential for maintaining the classification capability. If $C$ is the core, then (7). Where $R_i$ represents all possible reducts of the attribute set $A$, it is represented as $RED(A)$. This means that every reduct must include the attributes in the core. In summary, while a reduct provides a minimal subset of attributes sufficient for classification, the core identifies those attributes that are indispensable across all possible reducts.

$$CORE(A) = \bigcap_{R_i} R_i \in RED(A) \tag{7}$$

### 2.4.3. Neighbourhood Rough Set Theory

Neighbourhood Rough Set Theory (NRST) is an extension of classical rough set theory that incorporates the concept of "neighbourhood" to handle continuous and numerical data without the need for discretization. It defines neighbourhoods based on distance metrics and uses them to approximate decision classes, enabling feature selection, classification, and noise reduction. In financial applications such as stock price prediction, credit scoring, fraud detection, and portfolio optimization NRST can be used to identify relevant features or patterns from complex and noisy datasets. However, despite its strengths, NRST has several limitations when applied to real-world financial data, which we will now explore in detail The performance of NRS is highly dependent on the choice of parameters like the neighbourhood radius ($\varepsilon$) or the number of nearest neighbours($k$). Inappropriate values can lead to overfitting or under fitting. Too small ε results in too few neighbours, resulting in unstable approximations. Too large ε leads to the inclusion of irrelevant samples, resulting in a loss of discriminatory power.

For example, in credit risk assessment, choosing an inappropriate $\varepsilon$ can cause high-risk applicants to be grouped with low-risk applicants, leading to poor decision-making.

We consider an information system $S = (U, A)$, where $U$ is a finite and non-empty set of samples $\{x_1, x_2, \ldots, x_n\}$, known as a universe. The set $A$ represents attributes (also called inputs, features, or variables) $\{a_1, a_2, \ldots, a_m\}$ that define the characteristics of the samples. In this context, $\langle U, A\rangle$ forms a decision table when $A = C \cup D$, where C, represents the set of condition attributes and and $D$ is the decision attribute. For an object $x_i \in U$ and a subset $q \subseteq C$, the neighbourhood $\delta_P(x_i)$ of $x_i$ in feature space P is defined as follows:

$$\delta_P(x_i) = \{x_j \in U \mid \Delta^P(x_i, x_j) \leq \delta\} = [x_i]_{\delta,P} \tag{8}$$

This definition helps us identify objects that are similar to $x_i$ based on their attribute values within a specified distance $\delta$. The notation $\Delta^P$ represents a distance function, which satisfies the following properties for any $x_i, x_j, x_k \in U$:

(1) $\Delta^P(x_i, x_j) \geq 0$, (The distance is always non-negative.)
(2) $\Delta^P(x_i, x_j) = 0$ if and only if $x_i = x_j$ , (The distance between two identical points is zero.)
(3) $\Delta^P(x_i, x_j) = \Delta^P(x_j, x_i)$, (The distance between two points is symmetric.)
(4) $\Delta^P(x_i, x_j) + \Delta^P(x_j, k) \geq \Delta^P(x_i, x_k)$. (The triangle inequality holds.)

In this context, $\Delta^P$ defines a distance function, and the pair $\langle U, \Delta^P\rangle$ represents a metric space. For any two samples $x_i = (a_1(x_i), a_2(x_i), \ldots, a_m(x_i))$ a commonly used metric is the Euclidean distance, defined as follows:

$$\Delta^P(x_i, x_j) = \left(\sum_{k=1}^{N} |a(x_i) - a(x_j)|^2\right)^{\frac{1}{2}} \tag{9}$$

In this equation m represents the number of dimensions or features in the dataset, $a(x_i)$ *and* $a(x_j)$ denote the values of the $k-th$ feature for samples $x_i$ and $x_j$, respectively. The Euclidean distance measures the straight-line distance between two points in a multi-dimensional space, providing a quantitative measure of how similar or different the two samples are. It satisfies the properties of a metric, including non-negativity, identity of indiscernible, symmetry, and the triangle inequality. Thus, $(U, \Delta^P)$ forms a valid metric space.

### 2.4.4. Lower and Upper Neighbourhood Approximation

In the hybrid decision system $S = (U, C)$, where $q \subseteq C$ and $X \subseteq U$, $N_q$ denotes a neighborhood relation. The lower and upper approximations of $X$ with respect to the attribute set $q$ are defined by the following formulas [38]:

$$\underline{N_q}(\mathrm{X})=\{x_i \in U | [x_i]_{\delta,q} \subseteq X\} \quad (10)$$

$$\overline{N}_q(\mathrm{X})=\{x_i \in U | [x_i]_{\delta,q} \cap X \neq \emptyset\} \quad (11)$$

Obviously, $\underline{N_q}(\mathrm{X}) \subseteq X \subseteq \overline{N}_q(\mathrm{X})$. The boundary region of $X$ is defined as follows:

$$BND_q(\mathrm{X}) = \overline{N}_q(\mathrm{X}) - \underline{N_q}(\mathrm{X}) \quad (12)$$

Typically, the partition $U/d$ induces $X$, indicating a specific class. The approximations described above provide a method to characterize the specific class $X$ based on neighbourhood sets. The positive and negative regions of $X$ with respect to $q$ are defined as:

$$POS_q(\mathrm{X}) = \underline{N_q}(\mathrm{X}) \quad (13)$$

$$NEG_q(\mathrm{X}) = U - \overline{N}_q(\mathrm{X}) \quad (14)$$

### 2.4.5. Example

The hybrid decision system of a Portuguese banking institution consists of nominal and numerical attributes, which are presented in Table 1 and Table 2, respectively. Direct marketing campaigns serve as the data source for the banking system. The dataset includes 20 features and 54,211 customer records. In Tables 3 and 4, the data types, their descriptions, and corresponding categorical values are presented. The *"Selector"* serves as a class label to categorize groups $(0\ or\ 1)$. The campaigns employed phone calls, and in many instances, multiple calls were necessary for a single client to determine whether a product was subscribed $(1)$ or not $(0)$. We analyzed the information in Table 1 and Table 2 using neighborhood rough set theory $(\delta = 0.2)$. Clearly, the conditional attributes in the aforementioned tables consist of mixed data. Therefore, the decision system is defined as a hybrid $HDS = (U, V, C \cup d, f)$, where $C$ represents a combination of nominal and numerical data.

**Step1:** The numerical values in the decision table are normalized to a range between $0$ and $1$.

**Step2:** By applying Equations (10) and (11), we obtain the following values:

$$[x_1]_{\delta,C} = \{x_1\}, \quad [x_2]_{\delta,C} = \{x_2\},$$

$$[x_3]_{\delta,C} = \{x_3\}, \quad [x_4]_{\delta,C} = \{x_4\},$$

$$[x_5]_{\delta,C} = \{x_5, x_6\}, \quad [x_6]_{\delta,C} = \{x_5, x_6\},$$

$$[x_7]_{\delta,C} = \{x_7\}, \quad [x_8]_{\delta,C} = \{x_8\},$$

$$[x_9]_{\delta,C} = \{x_9\}, \quad [x_{10}]_{\delta,C} = \{x_{10}\}.$$

**Step3:** We compute the approximations of the decision classes using Equations (12) and (13). Additionally, we can divide the decision attribute into two subsets based on equivalence relations using the decision classes:

$U/d = \{\{x_1, x_2, x_3, x_4, x_6, x_8, x_9\}, \{x_5, x_7, x_{10}\}\},$

$D_1 = \{x_1, x_2, x_3, x_4, x_6, x_8, x_9\},$

$D_2 = \{x_5, x_7, x_{10}\},$

$\underline{N_X}(D_1) = \{x_1, x_2, x_3, x_4, x_8, x_9\},$

$\overline{N_X}(D_1) = \{x_1, x_2, x_3, x_4, x_6, x_8, x_9\}$

$\underline{N_X}(D_2) = \{x_7, x_{10}\},$

$\overline{N_X}(D_2) = \{x_5, x_6, x_7, x_{10}\}.$

**Step4:** The degree of dependency of the decision classes can be calculated using Equation (5):

$$\gamma_C(D) = \frac{|\{x_1, x_2, x_3, x_4, x_7, x_8, x_9, x_{10}\}|}{|\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}|} = 0.8$$

**Table 1:** Sample of Portuguese banking institution nominal attributes.

| U | Job | Marital | Education | Default | Housing | Loan | Contact | Month | Day | Pout come | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | blue-collar | married | basic.4y | unknown | yes | no | telephone | aug | tue | non-existent | 1 |
| $x_2$ | housemaid | divorced | degree | no | yes | yes | cellular | nov | thu | non-existent | 1 |
| $x_3$ | admin. | married | high. School | no | no | yes | cellular | aug | mon | success | 1 |
| $x_4$ | housemaid | divorced | course | no | yes | no | cellular | nov | mon | success | 1 |
| $x_5$ | technician | married | degree | no | yes | yes | cellular | may | fri | non-existent | 0 |
| $x_6$ | retired | married | degree | no | yes | no | cellular | mar | fri | non-existent | 1 |
| $x_7$ | management | single | basic.4y | no | no | no | telephone | may | mon | non-existent | 0 |
| $x_8$ | services | married | high.School | unknown | yes | no | telephone | may | mon | non-existent | 1 |
| $x_9$ | self-employed | divorced | high.School | no | no | no | cellular | sep | tue | success | 1 |
| $x_{10}$ | admin. | divorced | high.School | no | no | no | telephone | jul | mon | non-existent | 0 |

**Table 2:** Sample of Portuguese banking institution numeric attributes.

| U | Age | Duration | Campaign | P-days | Previous | Emp.var.rate | Cons.price.idx | Cons.conf.idx | month rate | Nr.employed | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 53 | 1186 | 4 | 999 | 0 | 1.4 | 93.444 | -36.1 | 4.968 | 5228.1 | 1 |
| $x_2$ | 54 | 653 | 1 | 999 | 0 | -0.1 | 93.2 | -42 | 4.076 | 5195.8 | 1 |
| $x_3$ | 31 | 155 | 2 | 4 | 1 | -2.9 | 92.201 | -31.4 | 0.884 | 5076.2 | 1 |
| $x_4$ | 67 | 655 | 2 | 5 | 5 | -1.1 | 94.767 | -50.8 | 1.039 | 4963.6 | 1 |
| $x_5$ | 41 | 170 | 4 | 999 | 0 | -1.8 | 92.893 | -46.2 | 1.313 | 5099.1 | 0 |
| $x_6$ | 73 | 179 | 1 | 999 | 0 | -1.8 | 92.843 | -50 | 1.531 | 5099.1 | 1 |
| $x_7$ | 32 | 73 | 7 | 999 | 0 | 1.1 | 93.994 | -36.4 | 4.858 | 5191 | 0 |
| $x_8$ | 41 | 679 | 2 | 999 | 0 | 1.1 | 93.994 | -36.4 | 4.857 | 5191 | 1 |
| $x_9$ | 39 | 261 | 1 | 3 | 1 | -3.4 | 92.379 | -29.8 | 0.788 | 5017.5 | 1 |
| $x_{10}$ | 48 | 352 | 2 | 999 | 0 | 1.4 | 93.918 | -42.7 | 4.96 | 5228.1 | 0 |

**Table 3:** Features Description of The Bank Marketing Financial Data Set

| No | Attributes | Data Type |
|---|---|---|
| 1 | Job | admin., blue-collar, entrepreneur, housemaid, management, retired, self-employed services, student, technician, unemployed, unknown |
| 2 | Marital | divorced, married, single, unknown |
| 3 | Education | basic.4y, basic.6y, basic.9y, high. School, illiterate professional. Course, university. Degree, unknown |
| 4 | Default | no, yes, unknown |
| 5 | Housing | no, yes, unknown |
| 6 | Loan | no, yes, unknown |
| 7 | Contact | cellular, telephone |
| 8 | Month | Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec |
| 9 | Day | Mon, Tue, Wed, Thu, Fri |
| 10 | Pout come | failure, nonexistent, success |

**Table 4:** Details of Impute Variables of Portuguese Bank Datase

| No | Attributes($a_i$) | Attribute Description | Data Type |
|---|---|---|---|
| 1 | Age | What is the customer's age? | Numeric |
| 2 | job | What is the customer's business status? | Nominal |
| 3 | Marital | What is the customer's marital status? | Nominal |
| 4 | Education | What is the customer's educational status? | Nominal |
| 5 | Default | What is the customer's credit debt status? | Nominal |
| 6 | Housing | What is the customer's real estate debt status? | Nominal |
| 7 | Loan | What is the customer's personal debt status? | Nominal |
| 8 | Contact | What is the customer's type of relationship? | Nominal |
| 9 | Month | When was the customer's last month of interview? | Numeric |
| 10 | Day | When was the last day of interview? | Numeric |
| 11 | Duration | How long was the last contact? | Numeric |
| 12 | Campaign | How many customers followed up during the campaign? | Numeric |
| 13 | P-days | How many times the customer called since the previous campaign | Numeric |
| 14 | Previous | How many times the customer been called before the campaign? | Numeric |
| 15 | Pout come | When did the previous marketing campaign end? | Nominal |
| 16 | Emp.var.rate | Employment Change Rate - Quarterly Index | Numeric |
| 17 | Cons.price.idx | Consumer Price Index - Monthly Index | Numeric |
| 18 | Cons.conf.idx | Consumer Confidence Index - Monthly Index | Numeric |
| 19 | month rate | European Interbank Offered Rate 3 Months - Daily Index | Numeric |
| 20 | Nr.employed | Number of Employees - Quarterly Index | Numeric |
| 21 | Customer | Has the customer registered a term deposit? | Binary |

## 3 Proposed Method

As mentioned earlier, formal Rough Set Theory (RST) can be used to reduce the dimensionality of data sets. This method can also serve as a pre-processing step for selecting the modelling approach to learn from the data. One of the main advantages of RST is its ability to provide abstraction, allowing it to be combined with various mathematical structures to achieve satisfactory results.

In this section, we introduce a method for reducing attributes by applying distance in the boundary region of neighbourhood rough set theory. Many existing rough-set-based feature selection methods rely on information obtained from the lower approximation of a set to minimize data. These methods are often seen as reliable because they focus on the certainty represented by the lower approximation, making them crucial for scientific analysis. However, while these approaches are generally effective, they tend to overlook the information found in the boundary or uncertainty region.

Additionally, some methods utilize the upper approximation. This means they evaluate the upper approximation as a whole, rather than treating the boundary region and lower approximation as distinct entities. To feature selection subsets, our proposed method considers both the lower and upper approximations of the information within the boundary region. As a result, we expect that the subset chosen through this method will be smaller than what would be obtained from using either the upper or lower approximation alone.

### 3.1. Distance Metric

Machine learning algorithms are widely used for data classification in various real-world scenarios. Selecting an efficient distance metric is crucial in this context. Distance metrics can enhance clustering and classification by identifying similarities between data and improving information retrieval. For mixed data types, rather than calculating distances separately for nominal and numerical variables, a single formula can be employed. In a hybrid decision system represented as $S = (U, V, C \cup D, f)$), $C$ denotes a set of conditional attributes that consists of two parts: $C_n$ (numerical attributes) and $C_m$ (nominal attributes). Thus, $C = C_n \cup C_m$. Let $K$ be a subset of $C$, meaning $K \subseteq C$, where , $K = K_n \cup K_m, K_n \subseteq C_n, K_m \subseteq C_m$. The neighbourhood relation R for the attribute set K is defined as:

$$R_K = \{(x_i, x_j) \in U \times U | (\Delta_{i,j})^K \leq \delta, 0 \leq \delta \leq 1\} \tag{15}$$

In this formula, $\Delta_{i,j}^2$ calculates as follows:

$$(\Delta_{i,j}^2)^K = \frac{\left(\sum_{l=1}^{|K_n|} \chi_l \left|\hat{f}_{al}(x_i) - \hat{f}_{al}(x_j)\right|^2 + \sum_{r=1}^{K_m} \chi_r \left(\varphi(x_i, x_j, a_r)\right)\right)}{\sum_{i=1}^{|n+m|} \chi_i} \quad (\forall a_l \in K_n, \forall a_r \in K_m) \tag{16}$$

This expression evaluates the distance between objects in numerical data as $\sum_{k=1}^{|K_n|} \chi_l \left|\hat{f}_{al}(x_i) - \hat{f}_{al}(x_j)\right|^2$ and the distance in nominal data as$\sum_{r=1}^{m} \varphi(x_i, x_j, a_r)$. According to the equation, if objects $i$ and $j$ share the same nominal value, the distance is 0; otherwise, it is 1.

When features are on different scales, this creates a challenge. Before calculating distances for numerical data, we must ensure the data is normalized; otherwise, one feature can dominate others. A common normalization method is min-max normalization, which converts the maximum and minimum values of a feature to 0 and 1, respectively, with other values falling between 0 and 1. The min-max

normalization for numerical data is defined as:

$$\hat{f}_{al} = \frac{f_{al}(x_i) - min(V_{al})}{max(V_{al}) - min(V_{al})} \tag{17}$$

## 3.2. Feature Selection Algorithm

Feature selection is a combinatorial optimization problem that plays a pivotal role in the field of data mining. This process significantly enhances the performance of learning algorithms by removing irrelevant and redundant features. In essence, feature selection entails the identification of a subset of features from the original feature set, thereby facilitating the extraction of patterns within a dataset and optimizing performance according to predefined objectives and criteria. The rough set theory approach to feature selection focuses on identifying minimal attribute sets and utilizes meta-heuristic algorithms to construct high-quality classifiers based on the selected features.

Let $O = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$ denote a collection of training samples, where $x_i \in U$ and $x_i = (a_1(x_i), a_2(x_i), \ldots, a_m(x_i))$ represents an $m$-dimensional feature vector, while $y_i \in \{0,1\}$ indicates the corresponding class label, with $n$ signifying the total number of samples. Define $X^+ = \{x_i | y_i = 1\}$ as the set of samples belonging to the majority class and $X^- = \{x_i | y_i = 0\}$ as those belonging to the minority class. The notation $\underline{N}(X^+)$ represents the samples from the majority class $X^+$, while $\overline{N}(X^+)$ includes samples from the minority class $X^-$, collectively forming the positive region. Additionally, Additionally $\underline{\Delta}(X^+)$ and $\bar{\Delta}(X^+)$ denote the boundary regions.

In this context, if two elements are neighbours, their neighbourhood structure remains preserved following feature selection, ensuring that the positive region remains invariant. Conversely, elements that are not initially neighbours may become neighbours as a consequence of the feature selection process. The objective is to identify the factor $\chi = (\chi_1, \chi_2, \ldots, \chi_m)$, where $\chi_i \in \{0,1\}$, which selects the optimal subset of features for classification purposes.

$$Min\ f = \sum_{i=1}^{m} \chi_i$$

Subject to:

$$\begin{cases} \Delta_{i,j}(\chi_1, \chi_2, \ldots, \chi_m) > 0, & (x_i, x_j) \in NEG(X^+) \times \overline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \ldots, \chi_m) > \delta, & (x_i, x_j) \in NEG(X^+) \times \underline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \ldots, \chi_m) > \delta, & (x_i, x_j) \in NEG(X^+) \times \underline{N}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \ldots, \chi_m) > 0, & (x_i, x_j) \in \underline{N}(X^+) \times \underline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \ldots, \chi_m) > \delta, & (x_i, x_j) \in \underline{N}(X^+) \times \overline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \ldots, \chi_m) > \delta, & (x_i, x_j) \in \underline{N}(X^+) \times NEG(X^+) \\ \chi_i \in \{0,1\} \end{cases} \tag{18}$$

This optimization problem aims to minimize the selection of features $f = \sum_{i=1}^{m} \chi_i$ while adhering to specific constraints that ensure effective separation between positive and negative classes in the dataset. The constraints dictate that the pairwise distances $\Delta_{i,j}(\chi_1, \chi_2, \ldots \chi_m)$ must satisfy certain conditions based on the classification of the samples. Specifically, if a sample $x_{\mathrm{i}}$ belongs to the negative class and another sample $x_j$ lies in the boundary or neighborhood of the positive class, their distance must be

greater than zero or exceed a predefined threshold $\delta$. These conditions ensure that selected features maintain $i-th$ a clear distinction between classes, ultimately enhancing the model's classification performance. The binary variable $\chi_i$ indicates whether a feature is included (1) or excluded (0), facilitating dimensionality reduction while preserving essential information for accurate predictions.

*Let* $\chi^{(1)} = (\chi_1^1, \chi_2^1, \ldots, \chi_m^1)$ be a solution to (20), then $R_1 = \{(a_1, \chi_1^1), \ldots, (a_m, \chi_m^1)\}$ is a reduct with minimum cardinality. If

$$\chi_j^1 = \begin{cases} 1, & j \in \Lambda_1 = \{l_{1,1}; l_{1,2}; \ldots; l_{1,m_1}\} \\ 0, & otherwise \end{cases}$$

Then $R_1 = \{a_{l_{1,1}}, a_{l_{1,2}}, \ldots, a_{l_{1,m_1}}\}$. We represent the product of non-zero component of $\chi^{(1)}$ by $\Pi^{(1)}$ as follows:

Considering $\Lambda_1$,

$$\Pi^{(1)} = \chi_{l_{1,1}}^1 \times \chi_{l_{1,2}}^1 \times \ldots \times \chi_{l_{1,m_1}}^1 \tag{19}$$

There are several ways to solve constrained optimization problems, one of which is the Penalty Function method. In this approach, proposed solutions can violate the problem's constraints, but each violation incurs a penalty based on its severity. This penalty affects the quality of the solution by altering the objective function value. For example, in a minimization problem, the penalty function increases the objective function, making the solution worse. While there are many ways to define a penalty function, certain key principles should guide its design. Based on the issues discussed and Example 2.4.5, we can draw the following conclusions:

$$C_1' = \{x_1, x_2, x_3, x_4, x_5, x_8, x_9\}, \qquad C_2' = \{x_7, x_{10}\}, \qquad C_1'' = \{x_6\}, \qquad C_2'' = \{x_5\}.$$

Equation (20) can be optimized using these constraints. The optimization method used here is similar to the feature selection approach discussed in this article. Various feature selection methods have been proposed, including heuristic methods for large datasets, which have proven to be effective. Meta-heuristic optimization algorithms can be slightly modified for different optimization problems.

With the increasing complexity of real-world issues and the need for quick solutions, traditional methods often fall short, leading to a rise in random algorithms. As a result, the use of heuristic and meta-heuristic algorithms has grown significantly over the past few decades. Unlike classical methods, heuristic search methods explore the search space in parallel and rely on a single fitness function to guide their search, leveraging swarm intelligence. Examples of these methods include the bird population algorithm, genetic algorithm, and firefly algorithm. Meta-heuristic algorithms can be categorized into population-based and path-based methods. The genetic algorithm discussed in this paper utilizes a set of strings, and its basic concepts will be explained further in the next sections. Many meta-heuristics employ stochastic optimization, meaning that the solutions found depend on randomly generated variables. In combinatorial optimization, meta- heuristics can efficiently find good solutions by exploring a large set of feasible options, often requiring less computational effort than traditional optimization algorithms or simple heuristics.

### 3.2.1. Genetic Algorithm

Genetic Algorithms (GAs) are optimization techniques inspired by the process of natural selection. They are used to solve complex problems by mimicking the evolutionary processes that occur in nature. The main components of a GA include a population of potential solutions, a fitness function to evaluate

these solutions, and operators such as selection, crossover, and mutation. GAs are particularly useful for optimization problems where traditional methods may struggle, as they explore a wide solution space and can escape local optima [19]. Here is the pseudocode for the genetic algorithm:

Alghorithm1: Pseudo-code of Genetic Algorithm

```
1. Initialize population with random individuals
2. Evaluate fitness of each individual in the population
3. Repeat until stopping condition is met:
    a. Select parents from the current population based on fitness
    b. Generate offspring through crossover and mutation
    c. Evaluate fitness of offspring
    d. Replace the old population with the new offspring
4. Return the best solution found
```

Genetic algorithms enable efficient exploration of diverse solution spaces by starting with an initial population of candidate features. Each feature is evaluated using a fitness function, and through iterative processes like selection, crossover, and mutation, new populations are generated that converge towards optimal solutions. By applying the genetic algorithm to Equation 20, we can obtain the reduct set with the smallest cardinality.

The time complexity of a Genetic Algorithm (GA) can be defined using the following parameters:

• N: Population size

• G: Number of generations

• D: Number of variables (dimensions)

• M: Number of inequality constraints

in each generation, every individual (solution vector of size $D$) must be evaluated. This evaluation includes checking all $M$ constraints. Additionally, basic operations such as crossover, mutation, and selection are proportional to $N$ and $D$.

Thus, the time complexity per generation is approximately: $O(N \cdot (M + D))$, Since this process repeats for $G$ generations, the overall time complexity is: $O(G \cdot N \cdot (M + D))$ GAs maintain population diversity while directing the search towards promising areas, allowing for the discovery of complex feature interactions often overlooked by traditional optimization methods. When combined with numerical optimization techniques, GAs enhance the fine-tuning of features, ensuring their individual and collective effectiveness in improving the proposed method. The following defines the constrained numerical optimization problem:

$$\begin{aligned} &Minimize\ f(x) \quad x \in R^n \\ &Subject\ to: \\ &\begin{cases} h_i(x) = 0, & i = 1,2,\dots,m; \\ g_i(x) \le 0, & i = m+1,2,\dots,p. \end{cases} \end{aligned} \tag{20}$$

In this formulation, $h_i(x)$represents equality constraints, while $g_i(x)$ denotes inequality constraints. The feasible region is defined by the set of vectors $x$ that satisfy all the constraints $h_i(x) = 0$ and $g_i(x) \le 0$. A vector that meets all these constraints is considered part of the feasible region.

In constrained optimization problems, we often seek to optimize a given objective function while satisfying certain constraints. However, handling constraints directly can be challenging. One effective approach to address this issue is the use of penalty functions. A penalty function is a technique that transforms a constrained optimization problem into an unconstrained one by incorporating the constraints into the objective function. The basic idea is to add a penalty term to the objective function that increases as the solution violates the constraints. This way, feasible solutions (those that satisfy the constraints) will have lower penalty values compared to infeasible solutions. [8]. Using this approach, we transform a constrained problem into a non-constrained problem as follows:

$$F(X) = \begin{cases} f(x) & x \in feasible\ region \\ f(x) + penalty(x) & x \notin feasible\ region 0 \end{cases} \tag{21}$$

The function $F(X)$ is an objective function, it can be divided into two scenarios: If $x$ is within the feasible region, meaning all constraints are satisfied, then the objective function is simply $f(x)$. This indicates that we only focus on the original objective function value. If $x$ is outside the feasible region, meaning one or more constraints are violated, then the objective function becomes $f(x)$ + penalty(x). Here, penalty$(x)$is an additional term that represents a penalty for violating constraints. This penalty is designed to increase as the violation worsens. Ultimately, if a point is neither in the feasible region nor has a defined penalty, the function value will be 0. This structure helps identify infeasible points during the optimization process and guides the search towards feasible solutions.

***Proposition:***

Let $R_i = \{a_{l_{i,1}}, a_{l_{i,2}}, \dots, a_{l_{i,m_i}}\}, i = 1,2, \dots, t$ are reducts to given information system such that $|R_1| \leq |R_2| \leq \cdots \leq |R_t|$ due to Equation (21) define $\Pi^{(2)}, \dots, \Pi^{(t)}$. Use a big M multiplier to reformulate (20) as follows:

$$Min\ f = \sum_{i=1}^{m} \chi_i + M \sum_{i=1}^{t} \Pi^{(j)}$$

Subject to:

$$\begin{cases} \Delta_{i,j}(\chi_1, \chi_2, \dots, \chi_m) > 0, & (x_i, x_j) \in NEG(X^+) \times \overline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \dots, \chi_m) > \delta, & (x_i, x_j) \in NEG(X^+) \times \underline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \dots, \chi_m) > \delta, & (x_i, x_j) \in NEG(X^+) \times \underline{N}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \dots, \chi_m) > 0, & (x_i, x_j) \in \underline{N}(X^+) \times \underline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \dots, \chi_m) > \delta, & (x_i, x_j) \in \underline{N}(X^+) \times \overline{\Delta}(X^+) \\ \Delta_{i,j}(\chi_1, \chi_2, \dots, \chi_m) > \delta, & (x_i, x_j) \in \underline{N}(X^+) \times NEG(X^+) \\ \chi_i \in \{0,1\} \end{cases} \tag{22}$$

Then this optimisation problem gives reduct $R_{t+1}$such that $|R_t| \leq |R_{t+1}|$.

For example, referring to Table 6, we see that $\chi_1 = \{\chi_1, \chi_3, \chi_7, \chi_{12}, \chi_{14}\}$ is one of the responses in Equation (20). Using Equation (24) under the same constraints, other reducts can be identified. Table 6 shows the various reducts obtained by the proposed method for the Bank data set. The smallest reduct listed includes attributes $1, 3, 7, 12, and\ 14$. Consequently, $\chi_1 = \chi_3 = \chi_7 = \chi_{12} = \chi_{14} = 1$, with all other attributes set to 0. The first two reducts each contain five attributes, indicating they are minimal subsets that retain sufficient information for classification. The third and fourth reducts have six attributes, possibly capturing more complex relationships within the data. The fifth reduct includes seven

attributes, suggesting a broader coverage but with potential redundancy. As shown in Table 6, the cardinalities of the sets are as follows: $R_1 = \{a_1, a_3, a_7, a_{12}, a_{14}\}$ has 5 elements, $R_2 = \{a_2, a_9, a_{11}, a_{13}, a_{14}\}$ has 5 elements, $R_3 = \{a_1, a_6, a_{11}, a_{13}, a_{14}, a_{15}\}$ has 6 elements, $R_4 = \{a_3, a_7, a_{10}, a_{11}, a_{12}, a_{14}\}$ has 6 elements and $R_5 = \{a_1, a_3, a_4, a_6, a_7, a_9, a_{12}\}$ has 7 elements. Therefore, the cardinalities satisfy: $|R_1| \leq |R_2| \leq |R_3| \leq |R_4| \leq |R_5|$. Choosing the appropriate reduct significantly affects model performance. Using fewer attributes, as in the first two reducts, tends to produce simpler models that are less likely to overfit. In contrast, more complex reducts, such as the fifth, may capture additional nuances but can reduce interpretability and increase the risk of overfitting.

**Table 6:** The Selected Reducts of Bank Dataset By Proposed Method

| No | $\|\chi\|$ | Position of $\chi$ | Reducts |
|---|---|---|---|
| 1 | 5 | 1, 3, 7, 12, 14 | $R_1 = \{a_1, a_3, a_7, a_{12}, a_{14}\}$ |
| 2 | 5 | 2, 9, 11, 13, 14 | $R_2 = \{a_2, a_9, a_{11}, a_{13}, a_{14}\}$ |
| 3 | 6 | 1, 6, 11, 13, 14, 15 | $R_3 = \{a_1, a_6, a_{11}, a_{13}, a_{14}, a_{15}\}$ |
| 4 | 6 | 3, 7, 10, 11, 12, 14 | $R_4 = \{a_3, a_7, a_{10}, a_{11}, a_{12}, a_{14}\}$ |
| 5 | 7 | 1, 3, 4, 6, 7, 9, 12 | $R_5 = \{a_1, a_3, a_4, a_6, a_7, a_9, a_{12}\}$ |

# 4 Experiments and Evaluation

The proposed algorithm was tested on the standard banking dataset in Table 5, and the results were compared with several other feature selection methods. Subsequently, different classification methods were employed to examine the effectiveness of the proposed method. The datasets were downloaded from the $UCI$ [13] and the implementation of $ROSE2$ software of the rough set theory tool [18]. The importance of using datasets from the UCI Machine Learning Repository in financial data analysis lies in their diversity and quality. The data for this study comes from a Portuguese banking institution aiming to predict customers' subscription to a term deposit product. The dataset includes four sets: two main sets, one with 41,188 samples and 20 features, and another with 41,188 samples and 17 features and two smaller sets, each containing 10% of the samples to facilitate running more complex algorithms like SVM. This dataset, especially the full version used in this study with 20 features, is considered a standard resource for analyzing and modeling customer behavior in direct marketing. These datasets provide a solid foundation for testing algorithms, developing models, and validating results. They help researchers and analysts understand financial patterns, improve decision-making, and enhance predictive accuracy. Additionally, using standardized datasets allows for better comparison of methods and findings across studies.

## 4.1. Feature Selection Algorithm

The proposed method was tested against several feature reduction methods using a rough set and neighbourhood rough set. The first method is a model of neighbourhood rough set to examine the issue of heterogeneous feature subset selection. The classical rough set can be utilized to examine categorical features and here, the classical rough set was generalized using a neighbourhood relations and a neighbourhood rough set model was introduced [39].

Another method is to use harmony search for selection of feature for high dimensional imbalanced class data in this study, a new method for selecting feature was introduced known as $SYMON$ that relies on symmetrical uncertainty and harmony search. To weigh features as to their dependence on class

labels, this method relies on symmetrical uncertainty [39]. In addition, for class imbalance learning, a weighted rough set-based method was introduced. This paper, introduced weights into the classic rough set model, which balances distribution of class of a data set and develops a weighted rough set-based method that handles the class imbalance problem [27]. In addition, selection of feature for imbalanced data on the basis of neighbourhood rough set are other compared methods in this article. In addition, we examined the uncertainty of feature selection based on different parameters. A particle swarm optimization algorithm was employed to achieve the optimized parameters in the algorithm [10].

## 4.2. Classification Method

Classification is a fundamental supervised learning technique widely employed in various domains, including finance, healthcare, and marketing, to categorize data into predefined classes. The process involves training a model on labelled datasets to enable accurate predictions for unseen instances. Support vector machines, neural networks, regression, decision trees, etc. are among the classification algorithms used by researchers, which have specific applications based on the complexity and type of data. Evaluation metrics such as accuracy, sensitivity, specificity, and ROC are essential for assessing model performance, guiding us in selecting the most suitable approach for specific tasks. Overall, a comprehensive evaluation of feature selection algorithms is vital for enhancing classification accuracy and model interpretability. The proposed model is evaluated using various classification methods, as follows:

### 4.2.1. Evaluation Metric

One of the performance evaluation criteria of classification models is accuracy, which evaluates the correctly predicted sample relative to the total samples, the formula of which is as follows:

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |FP| + |FN| + |TN|}, \tag{23}$$

in which,

TP (True Positive): The number of data instances that were actually positive and the model correctly predicted as positive.

TN (True Negative): The number of data instances that were actually negative and the model correctly predicted as negative.

FP (False Positive): The number of data instances that were actually negative but the model incorrectly predicted as positive (Type I error).

$FN$ (False Negative): The number of data instances that were actually positive but the model incorrectly predicted as negative (Type II error).

The confusion matrix (Fig.1), shows the performance of the classification model with the values of $TP, FN, FP,$ and $TN$.

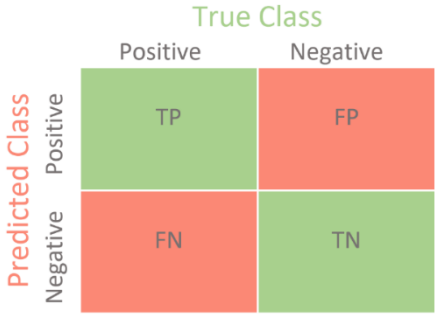


**Fig. 1:** Confusion Matrix: Performance of the Classification Model

Sensitivity is the true positive rate and specificity is the true negative rate, which respectively characterize the proportion of positive and negative tuples that are correctly identified.

$$\text{Sensitivity} = \frac{|TP|}{|TP| + |FP|} \tag{24}$$

$$\text{Specificity} = \frac{|TN|}{|TN| + |FP|} \tag{25}$$

The ROC curve is an essential instrument in evaluating classification models, providing valuable insights into their capabilities and limitations across different operating conditions. ROC curves are particularly useful in situations where class distribution is imbalanced, allowing practitioners to determine the model's robustness against different error types. By visualizing how many true positives can be achieved against false positives at various thresholds, the ROC curve aids in selecting an optimal cut-off point for predictions. We examined our financial dataset using the HFS, SYMON, RSFSAID and WRS feature selection methods to evaluate our proposed method. The results are shown in Table 7.

**Table 7:** The Selected Feature Financial Dataset By Various Feature Selection Algorithms, Compared With Proposed Method

| **Method** | **HFS [39]** | **SYMON [40]** | **WAR [27]** | **RSFSAID [7]** | **Proposed method** |
|---|---|---|---|---|---|
| Selected Feature | 1 2 3 7 13 14 15 16 18 19 (10) | 1 3 5 4 7 11 14 19 (8) | 1 2 4 5 6 7 8 10 11 12 19 (11) | 1 3 5 10 12 14 19 (7) | 3 7 10 11 12 14 (6) |

**Table 8:** AUC Using Various Classification Methods, Compared With Proposed Method

| **Method** | **HFS** | **SYMON** | **WAR** | **RSFSAID** | **Proposed method** |
|---|---|---|---|---|---|
| **Decision Tree(DT)** | 0.9213 | 0.9218 | 0.9376 | 0.9401 | 0.9499 |
| **J48** | 0.8343 | 0.8525 | 0.8012 | 0.8821 | 0.8901 |
| **ANN** | 0.8954 | 0.835 | 0.7921 | 0.9125 | 0.9209 |
| **SVM** | 0.7438 | 0.7321 | 0.6989 | 0.728 | 0.9660 |

Since the feature selection algorithms and feature evaluation procedures are not the same, the features selected by various feature selection methods differ. As mentioned, ROC and AUC are used to evaluate unbalanced data to verify the effectiveness of the feature selection algorithms used. The comparison of results for the method is reported in Table 8.

Figures 2 and 3 present the confusion matrices for the Bank dataset test data, comparing classifiers

with and without feature selection. In Figure 2, the matrices evaluate the performance of four algorithms: (a) Decision Tree, (b) J48, (c) Artificial Neural Network (ANN), and (d) Support Vector Machine (SVM), all without feature selection. Among these, J48 demonstrated the best overall balance, particularly in terms of recall for Class 2. The Decision Tree and ANN classifiers also performed well, each excelling in different areas. However, the SVM model faced challenges related to class imbalance. While it achieved an impressive accuracy of 99.3% for Class 2, it struggled with Class 1, resulting in a high misclassification rate of 64.4% due to a significant number of false negatives.

In Figure 3, the confusion matrices for the test dataset show the performance of four classifiers: (a) Decision Tree, (b) J48, (c) ANN, and (d) SVM. The rows indicate the predicted classes, while the columns represent the actual classes. Each cell displays the number of samples, the percentage contribution, and the class-wise accuracy for each category. Diagonal cells indicate correct classifications, while off-diagonal cells represent misclassifications. Feature selection plays a vital role in improving classification accuracy by removing irrelevant or redundant information. The matrices demonstrate that ANN and SVM outperform both Decision Tree and J48 classifiers, likely due to their ability to handle complex, nonlinear relationships more effectively. These results highlight the importance of both model selection and feature selection in analysing the Bank dataset for classification tasks.

The ROC curve visually represents the trade-off between the true positive rate and the false positive rate of a classifier. It is a useful tool for evaluating the performance of different models by illustrating their ability to distinguish between classes across various threshold settings. A curve closer to the top-left corner indicates better performance, reflecting higher sensitivity and specificity. The area under the ROC curve (AUC) quantifies overall accuracy, where values closer to 1 indicate a more effective classifier. ROC analysis is especially valuable for comparing models on imbalanced datasets, ensuring a balanced assessment beyond simple accuracy metrics [32]. Below is the ROC curve for the bank data without using feature selection and with feature selection applied.

Figure 4 presents the ROC (Receiver Operating Characteristic) curves for different classifiers applied to the Bank dataset without feature selection. Each subplot corresponds to a specific model: Decision Tree (a), J48 (b), Artificial Neural Network (ANN) (c), and Support Vector Machine (SVM) high classification performance. Specifically, the Decision Tree and J48 models (plots a and b) exhibit almost perfect separation between the classes, as indicated by the steep rise of their ROC curves and the proximity to the y-axis. This suggests that these models achieve high sensitivity with minimal false positive rates. The ANN (plot c) also demonstrates strong performance, with its ROC curves showing quick increases in the true positive rate.

However, there is a slightly more gradual ascent compared to the Decision Tree-based models, which may indicate marginally lower discrimination. In contrast, the SVM (plot d) demonstrates a noticeably less steep ROC curve, especially for Class 2, indicating somewhat lower overall classification performance. The curve for Class 2 in the SVM plot rises more slowly and does not reach the maximum true positive rate as rapidly as the other models, implying that the SVM may be less effective on this particular dataset without prior feature selection**.**

In summary, the Decision Tree and J48 classifiers outperform both ANN and SVM in terms of ROC characteristics on the raw Bank dataset, effectively distinguishing between classes with fewer false positives. This comparison highlights the importance of model selection when working with datasets that have not undergone feature selection.

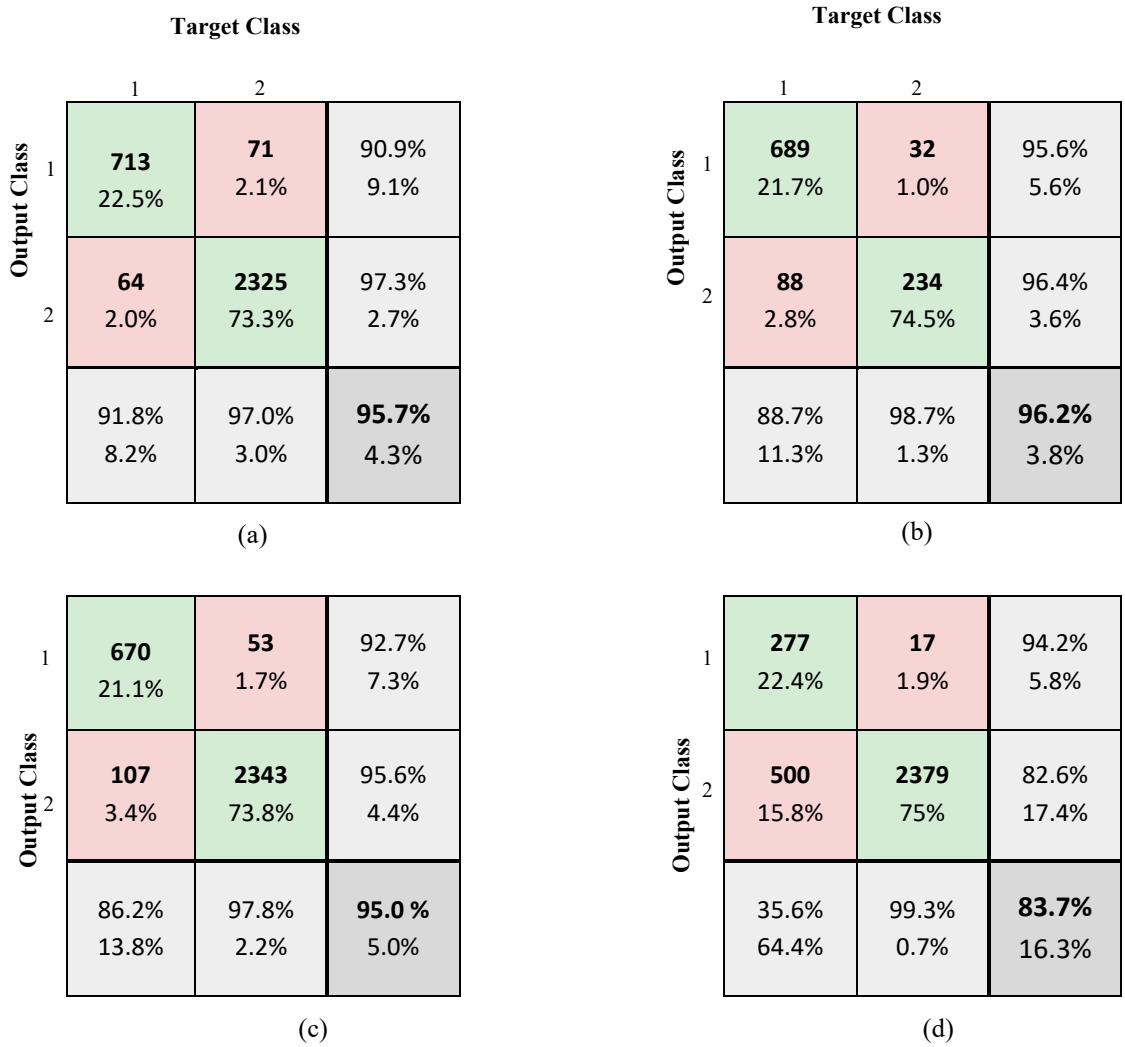


**Fig. 2:** Confusion matrix of Bank dataset without feature selection**.** (a). Confusion Matrix on Test Dataset (Decision Tree), (b). Confusion Matrix on Test Dataset (J48), (c). Confusion Matrix on Test Dataset (ANN), (d). Confusion Matrix on Test Dataset (SVM).

Figure 5 illustrates the ROC curves for various classifiers applied to the bank dataset following feature selection. These curves indicate the high performance of all classifiers, with their proximity to the top-left corner of the graph signifying a strong ability to identify positive instances while minimizing false positives. Distinct curves for classes 1 and 2 suggest that the models exhibit similar predictive capabilities for both classes. Additionally, the similarity in the curves across different classifiers implies that feature selection has contributed to the enhancement of the dataset's quality. The grey diagonal line in each subplot represents random performance (AUC = 0.5), and the positioning of the curves above this line confirms the effectiveness of the classifiers. Overall, the results demonstrate that feature selection plays a crucial role in improving classifier performance within the bank dataset. The AUC (Area Under the Curve) is a key metric for evaluating classification models, especially binary ones. It measures how well a model distinguishes between positive and negative classes by calculating the area under the ROC curve, which plots true positive rate versus false positive rate across different thresholds. Figure 6 compares AUC values of five feature selection methods (HFS, SYMON, WAR, RSFSAID,

and Proposed method) across four classifiers (Decision Tree, J48, ANN, SVM). The Proposed method consistently achieves the highest AUC, peaking at 0.965 with SVM, showing its effectiveness in selecting key features and boosting classifier performance. WAR performs the worst, especially with ANN and SVM. HFS, SYMON, and RSFSAID show moderate results but are outperformed by the proposed method. Decision Tree and J48 do well with most methods, while SVM benefits most from the proposed method, highlighting their strong combination. Overall, the proposed method significantly improves classification accuracy and is a robust choice for feature selection.


**Fig. 3:** Confusion matrix of Bank dataset with feature selection**.** (a). Confusion Matrix on Test Dataset(Decision Tree), (b). Confusion Matrix on Test Dataset (J48), (c). Confusion Matrix on Test Dataset (ANN), (d). Confusion Matrix on Test Dataset (SVM).

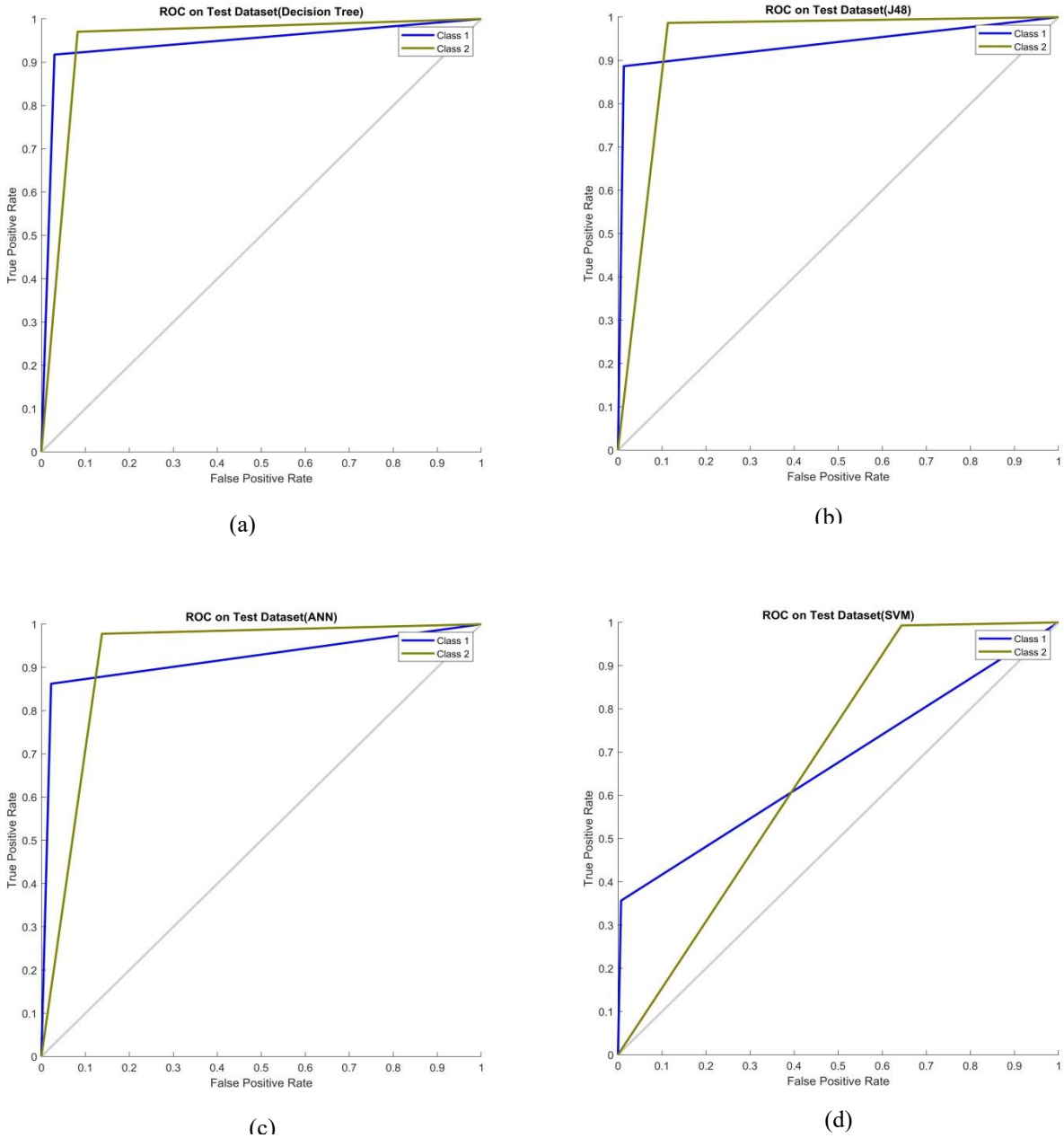


**Fig. 4:** ROC of Bank dataset without feature selection. (a). ROC on test dataset (Decision Tree), (b). ROC on test dataset (J48), ROC on test dataset (ANN), ROC on test dataset (SVM)

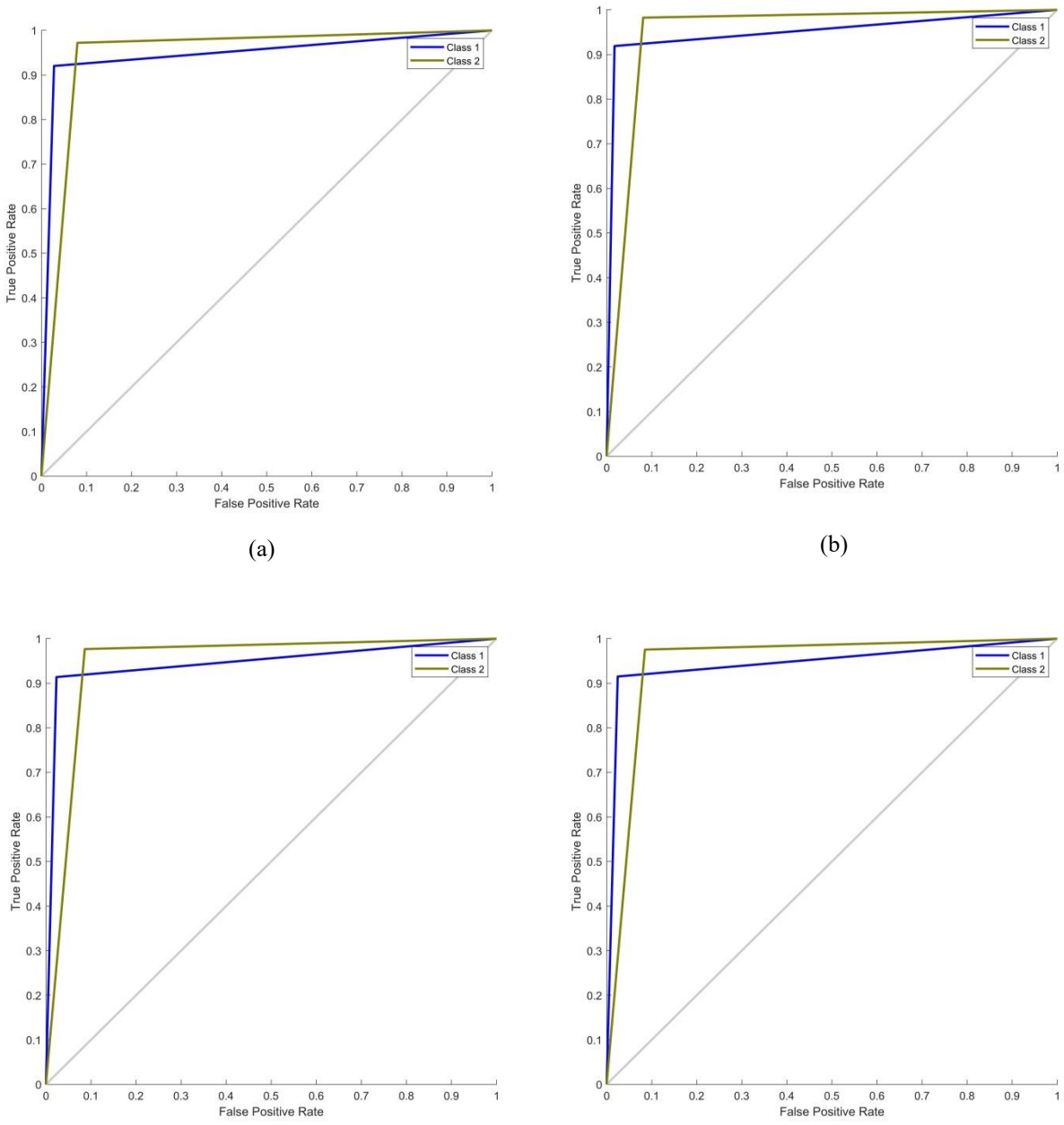


**Fig. 5:** ROC of Bank dataset with feature selection. (a). ROC on test dataset (Decision Tree), (b). ROC on test dataset (J48), ROC on test dataset (ANN), ROC on test dataset (SVM)

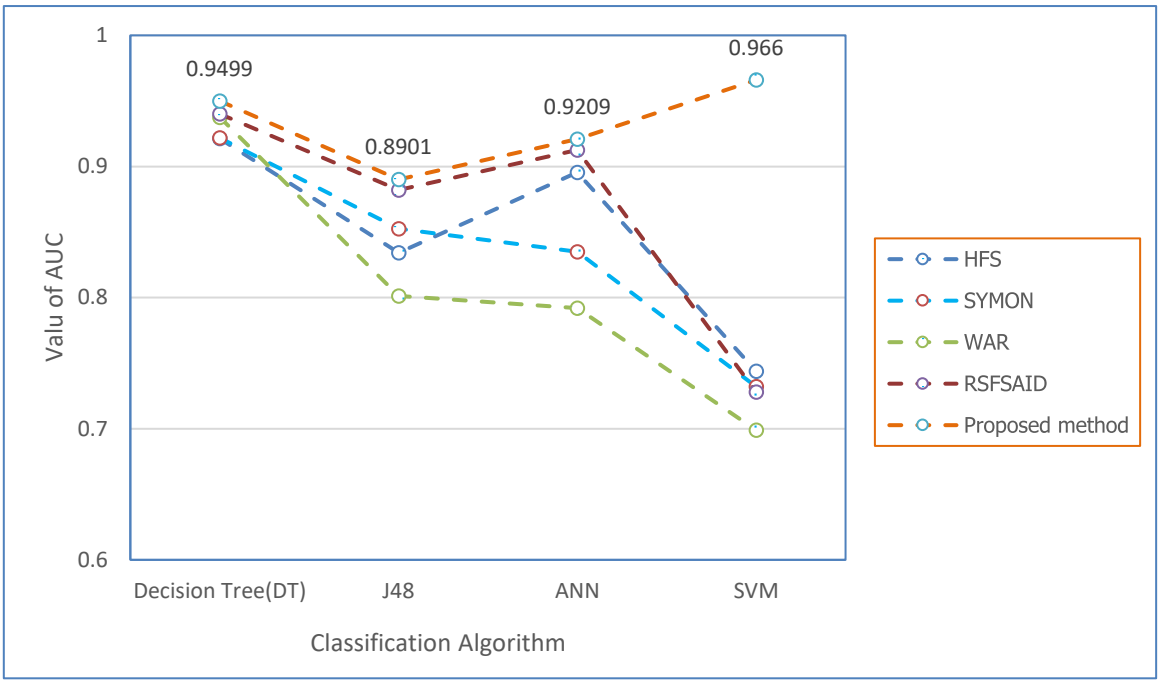


**Fig. 6:** Value of AUCs for Various Algoriths Based on Table 8.

## 5 Discussion and Conclusions

The costs of misclassifying minority class data in data science applications can be very high. This can lead to many problems when the data dimensions are large. The conventional approach to solving this problem is to select features to best predict minority class data. In this study, the problem of feature selection by using the neighbourhood roughest concept and the genetic algorithm was studied by presenting a new method for the selection of attributes. This method is based on elements of boundary region in neighbourhood rough set theory; the proposed method makes it possible to obtain fewer features at an acceptable time. The proposed algorithm was tested on several standard datasets. Various datasets (nominal, numerical, mix, and imbalance) were used to test the proposed method and experimental results show that this algorithm has reasonable accuracy compared to other methods. The number of features in the proposed method is significantly different from the original number of features. Feature selection plays a vital role in financial data analysis by reducing dimensionality, removing irrelevant or redundant variables, and enhancing model efficiency. This process improves prediction accuracy in tasks such as stock price forecasting, credit risk assessment, and fraud detection by concentrating on the most relevant factors. It also helps prevent overfitting, making models more generalizable to new data, and effectively addresses imbalanced classes often present in financial datasets. Moreover, narrowing down features increases model interpretability, allowing analysts to better understand key market drivers. On the other hand, the Neighborhood Rough Set Theory (NRST) provides valuable insights but comes with notable limitations. Its performance heavily depends on parameter settings like neighborhood size; improper tuning can reduce accuracy and generalizability. NRST may struggle with noisy or highly nonlinear financial data, where capturing complex patterns is challenging. Scalability issues emerge with large, high-dimensional datasets due to increased computational demands, hindering

real-time applications. Furthermore, NRST alone may not fully capture all aspects of financial risk and uncertainty, often requiring integration with other methods. In summary, while feature selection is essential for boosting model performance and interpretability in financial analysis, NRST should be applied with caution, considering its sensitivities and limitations. Combining NRST with complementary techniques can help overcome its drawbacks and lead to more robust financial models.

## References

[1] Abdel-Basset, M., Abdle-Fatah, L., Sangaiah, A.K., An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment, *Cluster Comput,* 2019; 22(4): 8319–8334. Doi.org/10.1007/s10586-018-1769-z.

[2] Ahmed, S., Ghosh, K.K., Singh, P.K., Geem, Z.W., Sarkar, R., Hybrid of Harmony Search Algorithm and Ring Theory-Based Evolutionary Algorithm for Feature Selection, *IEEE Access,* 2020; 8:102629–45. Doi: 10.1109/ACCESS.2020.2997005.

[3] Alazzam, H., Sharieh, A., Sabri, K.E., A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer, *Expert Syst Appl*, 2020; 148: 113249. Doi: 10.1016/j.eswa.2020.113249.

[4] Alavi, S. M, A., Khazravi. N., Evaluation and ranking of fuzzy sets under equivalence fuzzy relations as α−certainty and β−possibility, *Expert Systems with Applications,* 2024; 248:(123175): 0957-4174, Doi :10.1016/j.eswa.2024.123175.

[5] Aljarah, I., Faris, H., Mirjalili, S., Optimizing connection weights in neural networks using the whale optimization algorithm, *Soft Comput,* 2018: 22: 1433–7479. Doi: 10.1007/s00500-016-2442-1.

[6] Behrooz, S., Ghomi, R., Mehrazin, A., Shoorvarzi., M. Developing Financial Distress Prediction Models Based on Imbalanced Dataset , *Random Undersampling and Clustering Based Undersampling Approaches*, 2024; 9(3):737–62. Doi: 10.22034/amfa.2024.2189537.1689.

[7] Chen, H., Li, T., Fan, X., Luo, C., Feature selection for imbalanced data based on neighborhood rough sets, *Inf Sci (Ny,.* 2019: 483. Doi: 10.1016/j.ins.2019.01.073.

[8] Coello, C.A., Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput Methods Appl Mech Eng,* 2002; 191(11):1245–87. Doi:10.1016/S0045-7825 (01)00323-4.

[9] Dai, J., Hu, H., Wu, W.Z., Qian, Y., Huang, D., Maximal-Discernibility-Pair-Based Approach to Attribute Reduction in Fuzzy Rough Sets, *IEEE Trans Fuzzy Syst*, 2018; 26(4): 2174–87. Doi:101109/TFUZZ.2017.2768044.

[10] Dash, M., Liu, H., Feature selection for classification, *Intell Data Anal*, 1997; 1(1): 131–56. Doi:10.1016/ S1088-467X (97)00008-5

[11] Dhiman, G., Oliva, D., Kaur, A., Singh, K.K, Vimal, S., Sharma, A., et al., BEPO: A novel binary emperor penguin optimizer for automatic feature selection, *Knowledge-Based Syst*, 2021; 211: Doi:10.1016/ j.knosys. 2020.106560

[12] Dhiman, G., Kumar, V., Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, *Knowledge-Based Syst,* 2019; 165: 169–96. Doi: 10.1016/j.knosys.2018.11. 024.

[13] Dua, D., Graff, C., *{UCI} Machine Learning Repository.* 2017: Available from: http://archive.ics.uci.edu/ml. Doi: 10.24432/C56C7D

[14] Elango, S., Chandran, S., Mahalakshmi, S., Rough set-based feature selection for credit risk prediction using weight-adjusted boosting ensemble method, *Soft Comput,* 2020; 24. Gpi: 10.1007/s00500-020-05125-1.

[15] Eskandar, H., Sadollah, A., Bahreininejad, A., Hamdi, M., Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, *Comput Struct*, 2012; 110–111: 151–66. Doi:10.1016/j.compstruc.2012.07.010.

[16] Ghaemi, M., Feizi-Derakhshi, M.R., Forest optimization algorithm, *Expert Systems with Applications,* 2014; 41(15): 6676–6687. Doi: 10.1016/j.eswa.2014.03.034.

[17] Guoyin, W., Xiao, M., Haiyun Y., Monotonic uncertainty measures for attribute reduction in probabilistic rough set model, *International Journal of Approximate Reasoning,* 2015; 59: 41–67. Doi:10.10 16/j.ijar. 2015.01.002.

[18] Han, J., Kamber, M., Pei, J., Data Mining: Concepts and Techniques , *3rd ed. Waltham, Mass: Morgan Kaufmann Publishers;* 2012.

[19] Holland, J.H., Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, *MIT Press;* 1992.

[20] Hussien, A.G., Abualigah, L., Abu Zitar, R., Hashim, F.A., Amin, M., Saber, A., et al., Recent Advances in Harris Hawks Optimization: *A Comparative Study and Applications, Electronics,* 2022; 11(12): Doi:10.3390/electronics11121888.

[21] Hu, M., Tsang, E.C.C., Guo, Y., Chen, D., Xu, W., A novel approach to attribute reduction based on weighted neighborhood rough sets, *Knowledge-Based Systems,* 2021; 220: 106908. Doi:10.1016/ j.knosys. 2021.106908.

[22] Jensen, R., Tuson, A., Shen, Q., Finding rough and fuzzy-rough set reducts with SAT*, Inf Sci (Ny),* 2014; 255: 100–120. Doi: 10.1016/j.ins.2013.08.050.

[23] Kennedy, J., Eberhart, R., Particle swarm optimization. In: Proceedings of ICNN'95 - *International Conference on Neural Networks,*1995; 4: 1942–1948 Doi: 10.1109/ICNN.1995.488968.

[24] Khushaba, R.N., Al-Ani, A., AlSukker, A., Al-Jumaily, A., A Combined Ant Colony and Differential Evolution Feature Selection Algorithm*, Berlin, Heidelberg: Springer Berlin Heidelberg*, 2008; 1–12. Doi: 10.1007/978-3-540-87527-7-1.

[25] Ranjini, S.K.S., Murugan, S., Memory Based Hybrid Dragonfly Algorithm for Numerical Optimization Problems, *Expert Systems with Applications,* 2017; 83(C): 63–78. Doi: 10.1016/j.eswa.2017.04.031.

[26] Liang, B., Zhang, H., Lu, Z., Zhang, Z., Indistinguishable Element-Pair Attribute Reduction and Its Incremental Approach*, Mathematical Problems in Engineering*, 2022; 1–16. Doi: 10.1155/2022/4823216.

[27] Liu, J., Hu, Q., Yu, D., A weighted rough set based method developed for class imbalance learning, *Inf Sci (Ny)*, 2008; 178(4): 1235–1256. Doi: 10.1016/j.ins.2007.09.036.

[28] Liu, S., Zhang, J., Xiang, Y., Zhou, W., Xiang, D., A Study of Data Pre-processing Techniques for Imbalanced Biomedical Data Classification, *Int. J. Bioinformatics Res. Appl*, 2020; 16(3): 290–318. Doi:10.1504/ijbra.2020.109103.

[29] Mafarja, M., Mirjalili, S., Whale optimization approaches for wrapper feature selection, *Appl Soft Comput,* 2018; 62: 441–453. Doi: 10.1016/j.asoc.2017.11.001.

[30] Maksood, F., Achuthan, G., Analysis of Data Mining Techniques and its Applications*, Int J Comput Appl,* 2016; 140: 6–14. Doi: 10.5120/ijca2016909411.

[31] Murali, V., Fuzzy equivalence relations, *Fuzzy Sets Syst,* 1989; 30(2): 155–163. Doi: 10.1016/0165-0114(89)90017-1.

[32] Nahari, J., Qala, A., Rezaei, N., Aghdam, Y., Abdi, R., The Performance of Machine Learning Techniques in Detecting Financial Frauds, *Advances in Mathematical Finance & Applications Comparing,* 2024: 9(3): 1006–1023. Doi :10.71716/amfa.2024.22101813.

[33] Pandey, A. C., Kulhari, A., Mittal, H., Tripathi, A. K., Pal, R., Improved exponential cuckoo search method for sentiment analysis, *Multimedia Tools and Applications,* 2022; 82(16), 23979–24029. Doi:10.1007/s11042-022-14229-5.

[34] Pawlak, Z., Rough set approach to knowledge-based decision support, *Eur J Oper Res,* 1997; 99(1): 48–57. Doi: 10.1016/S0377-2217(96)00381-7.

[35] Pawlak, Z., Rough classification, *Int J Man Mach Stud,* 1984; 20(5): 469–483. Doi:10.1016/S0020-7373(84)80022-X.

[36] Pawlak, Z., Rough sets and intelligent data analysis, *Inf Sci (Ny),* 2002; 147(1): 1–12. Doi: 10.1016/S0020-0255(02)00197-4.

[37] Pieta, P., Szmuc, T., Kluza, K., Comparative Overview of Rough Set Toolkit Systems for Data Analysis, *MATEC Web Conf,* 2019; 252: 3019. Doi: 10.1051/matecconf/201925203019.

[38] Hu, Q., Yu, D., Liu, J., Wu, N., Neighborhood rough set based heterogeneous feature subset selection, *Inf Sci (Ny)*, 2008; 178(18): 3577–3594. Doi: 10.1016/j.ins.2008.05.024.

[39] Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., GSA: A Gravitational Search Algorithm, *Inf Sci (Ny)*, 2009; 179(13): 2232–2248. Doi: 10.1016/j.ins.2009.03.004.

[40] Sağlam, F., Sözen, M., Cengiz, M.A., Optimization Based Undersampling for Imbalanced Classes, *Adıyaman University Journal of Science*, 2021; 11(2): 385-409. Doi:10.37094/adyujsci.884120.

[41] Sepehri, A., Ghodrati, H., Jabbari, H., Panahian, H., Making Decision on Selection of Optimal Stock Portfolio Employing Meta Heuristic Algorithms for Multi-Objective Functions Subject to Real-Life Constraints, *Advances in Mathematical Finance & Applications,* 2023; 8(2): 645–666. Doi: 10.22034/ AMFA.2021. 1915292.1525.

[42] Shekhawat, S.S., Sharma, H., Kumar, S., Nayyar, A., Qureshi, B., bSSA: Binary Salp Swarm Algorithm With Hybrid Data Transformation for Feature Selection, *IEEE Access*, 2021; 9: 14867–14882. Doi:10.1109/ACCESS.2020.3047773

[43] Shareef, H., Ibrahim, A.A., Mutlag, A.H., Lightning Search Algorithm, *Applied Soft Computing*, 2015; 36(C): 315–333. Doi: 10.1016/j.asoc.2015.07.028.

[44] Wang, G.G., Deb, S., Cui, Z., Monarch Butterfly Optimization, *Neural Computing and Applications*, 2019; 31(7): 1995–2014. DOI: 10.1007/s00521-017-3210-z.

[45] Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R., Feature Selection based on Rough Sets and Particle Swarm Optimization, *Pattern Recognition Letters*, 2007; 28: 459–471. Doi: 10.1016/j.patrec.2006.09.003.

[46] Zhang, Y., Wang, Y., Research on Classification Model based on Neighborhood Rough Set and Evidence Theory, *Journal of Physics: Conference Series,* 202: 1746:012018. Doi: 10.1088/1742-6596/1746/1/012018.