

A rule-based machine learning model for financial fraud detection

Saiful Islam¹, Md. Mokammel Haque¹, Abu Naser Mohammad Rezaul Karim²

¹Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram, Bangladesh

²Department of Computer Science and Engineering, International Islamic University Chittagong, Chattogram, Bangladesh

Article Info

Article history:

Received May 2, 2023

Revised May 28, 2023

Accepted Jun 4, 2023

Keywords:

Data resampling

Fraud detection

Machine learning

Rule generation

Support confidence

ABSTRACT

Financial fraud is a growing problem that poses a significant threat to the banking industry, the government sector, and the public. In response, financial institutions must continuously improve their fraud detection systems. Although preventative and security precautions are implemented to reduce financial fraud, criminals are constantly adapting and devising new ways to evade fraud prevention systems. The classification of transactions as legitimate or fraudulent poses a significant challenge for existing classification models due to highly imbalanced datasets. This research aims to develop rules to detect fraud transactions that do not involve any resampling technique. The effectiveness of the rule-based model (RBM) is assessed using a variety of metrics such as accuracy, specificity, precision, recall, confusion matrix, Matthew's correlation coefficient (MCC), and receiver operating characteristic (ROC) values. The proposed rule-based model is compared to several existing machine learning models such as random forest (RF), decision tree (DT), multi-layer perceptron (MLP), k-nearest neighbor (KNN), naive Bayes (NB), and logistic regression (LR) using two benchmark datasets. The results of the experiment show that the proposed rule-based model beat the other methods, reaching accuracy and precision of 0.99 and 0.99, respectively.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Saiful Islam

Department of Computer Science and Engineering, Chittagong University of Engineering and Technology
Chittagong, Bangladesh

Email: engsaiful0@gmail.com

1. INTRODUCTION

The Oxford Dictionary describes fraud as an unjustified or criminal deception leading to monetary or personal advantage [1]. Fraud can occur in various financial industries, including banking, insurance, taxation, and corporations. Credit card fraud, tax evasion, financial statement fraud, money laundering, and other financial fraud are all rising. Fraud efforts have increased significantly in recent years, making fraud detection more critical than ever. Because of increased credit card use, there has been a constant increase in fraudulent transactions [2]. Asset misappropriation, corruption, and financial statement fraud are three categories of occupational fraud identified. In order to steal money, fraudulent transactions are frequently carried out using unlawful access to card information, including credit card numbers [3], email addresses, phone numbers [4], and many others. As the technology employed by the financial banking sector evolved during the last two decades, so did the fraud techniques used by criminals (European Payments Council 2019). Credit card fraud is now the second most prevalent sort of identity theft recorded as of this year, only following government documents and benefits fraud [5]. Fraud detection is critical with various high-impact applications in security, banking [6], health care [7], and review management. This research focuses on

financial statement fraud. Traditional fraud detection methods, such as manual detection, are costly, inaccurate, time-consuming, and ineffective [8]. Financial fraud is a broad term with many different definitions. Still, it can be described as the deliberate employment of illegal procedures or activities to obtain financial benefit [9]. According to a recent report, credit card fraud cost consumers around 27.85 billion dollars in losses in 2018, an increase of 16.2% over the 23.97 billion dollars lost in 2017. It is predicted to cost consumers 35 billion dollars by 2023 [10]. According to some estimates, the overall annual cost to the United States might surpass \$400 billion [9]. In contrast, a third study predicts that United Kingdom (UK) insurers lose 1.6 billion pounds each year owing to false claims. Financial fraud has far-reaching consequences for the industry, including supplying funds for illegal operations such as drug trafficking and organized crime [11]. Credit card fraud costs are typically borne by retailers responsible for shipping, chargeback, administrative charges, and losing consumer confidence due to a fraudulent purchase [12]. As a result, we can see the wide-ranging implications of fraud and the need to prevent it. As a result, financial institutions must prioritize the implementation of an automated fraud detection system.

The issue with machine learning is that there is a class imbalance when there are significantly more instances of one class of data (positive) than instances of another (negative). Numerous studies have been undertaken on the categorization difficulty of the unbalanced dataset. The problem of class imbalance is a significant concern in all current fraud detection models. If not addressed, these models may not be able to predict fraudulent transactions accurately. To mitigate this issue, many models require time-consuming re-sampling techniques during training. In light of this, we propose using a rule-based machine learning model to classify financial transactions as either fraudulent or non-fraudulent without resampling. This model is designed to identify patterns in the data using a set of decision rules, making it more interpretable and explainable than other machine learning models. This research aims to detect fraudulent financial transactions through a rule-based model that does not involve any re-sampling technique, which is a revolutionary idea in the realm of financial fraud detection in machine learning. Because this is the first time a rule-based model has been able to classify financial transactions without the need for data resampling accurately. This research makes the following contributions: i) we proposed a rule-based financial fraud detection model and ii) we apply the proposed rule-based financial fraud detection model to test it is effectiveness on two benchmark-skewed synthetic financial transaction datasets.

The rest of the paper is organized as follows. A summary of prior studies using machine learning (ML) to identify financial fraud is provided in section 2. Section 3 discusses the methodology of the study. The experimental data and analysis are presented in section 4. Finally, section 5 brings the research to a close.

2. RELATED WORK

Using various machine learning techniques such as supervised, semi-supervised, and unsupervised learning, researchers have created a number of models to automate financial fraud detection systems. Esenogho *et al.* [13] proposed a system that effectively detects credit card fraud by integrating a hybrid data resampling technique with a neural network ensemble classifier. The ensemble classifier in the adaptive boosting (AdaBoost) technique is created utilizing a long short-term memory (LSTM) neural network as the basis learner. Nguyen *et al.* [14] proposed a hybrid strategy utilizing CatBoost and deep learning. The key concept of the proposed model is user separation, in which consumers are divided into old and new users before applying CatBoost, and deep neural networks (DNNs) are applied to each group independently. When put into use, this model should be able to more precisely identify suspicious financial transactions and alert the appropriate authorities promptly to enable them to take the necessary action. Hashemi *et al.* [15] CatBoost and XGBoost were proposed as methods to improve the performance of the light gradient boosting machine (GBM) approach by taking the voting mechanism and weight-tuning as a pre-process for unbalanced input into account. Ileberi *et al.* [16] proposed a machine learning (ML) technique for detecting credit card fraud using real-world imbalanced datasets generated by European credit cards. To address the issue of class imbalance, they resampled the dataset using the synthetic minority oversampling technique. Synthetic minority oversampling technique (SMOTE). This system was evaluated using support vector machine (SVM), linear regression (LR), random forest (RF), extreme gradient boosting (XGBoost), decision tree (DT), and extra tree. To increase classification accuracy, these machine learning algorithms were integrated with the adaptive boosting (AdaBoost) approach. The Matthews correlation coefficient (MCC), the AUC, the recall, and the precision of the models were used to evaluate their performance (AUC). Taha and Malebary [17] suggested an intelligent approach for detecting credit card fraud (OLightGBM). The proposed method intelligently combines a Bayesian-based hyperparameter optimization technique to alter the parameters of a light gradient boosting machine (LightGBM).

Both association and classification rules are standard for rule-based modelling in machine learning and data science [18]. Numerous well-liked classification methods have been developed over the past few decades, including support vector machine (SVM), naive Bayes (NB), k-nearest neighbor (KNN), random forest (RF), logistic regression (LR), and genetic algorithm (GA) algorithms for feature selection [19] have been proposed. Bakhtiari *et al.* [20] provide ensemble learning techniques for identifying credit card fraud that incorporate gradient boosting (LightGBM and LiteMORT), and they combine these techniques by employing averaging techniques (simple and weighted averaging techniques) before being evaluated. By combining these approaches, error rates are decreased while efficiency and accuracy are improved. A unique representation learning (RL)-based network-based credit card fraud detection method was developed by Belle *et al.* [21], and it can help with fraud detection by avoiding manual feature engineering and directly taking transactional relationships into account. Salekshahrezaee *et al.* [22] used a dataset and four ensemble classifiers to investigate the effects of feature extraction and data samples on credit card fraud detection. They assessed the effectiveness of random under sampling (RUS), SMOTE, and SMOTE Tomek methods for data sampling as well as principal component analysis (PCA), convolutional autoencoder (CAE), and RUS methods for feature extraction. According to the results, the best performance for identifying credit card fraud was attained by combining RUS and CAE.

Fanai and Abbasimehr [23] introduced a two-stage method for identifying fraudulent transactions that makes use of representation learning with deep autoencoders and supervised deep learning algorithms. The technique improved the efficiency of deep learning-based classifiers, with classifiers trained on the Autoencoder's modified data set outperforming baseline classifiers trained on the original data in all performance measures. The deep autoencoder-based models outperformed those employing the dataset produced from PCA and the pre-existing models. Ahmad *et al.* [24] created a method for handling unbalanced data that involves under-sampling and clustering using fuzzy C-means to choose comparable fraud and normal examples with the same attributes. This strategy aims to maintain the integrity of the data feature while increasing accuracy and performance with different machine learning methods. Ni *et al.* [25] proposed a model for identifying credit card fraud that incorporates a spiral oversampling balancing technique (SOBT) and a method for boosting fraud attributes. In order to identify fraudulent cashback transactions in Indonesian e-commerce, Karunachandra *et al.* [26] employed machine learning. They used transaction data from a prominent e-commerce platform in the nation to train their model and employed supervised classification techniques like k-NN, CNN, and LSTM. For dealing with fraudulent cashback practices in the future, the report offers solutions. Lai *et al.* [27] developed a brand-new deep mixture model-based consumer fraud detection method called BTextCAN to spot fraud in the marketplace based on how a specific customer group views it. The suggested approach can mine consumer opinions and use their collective perspective to identify consumer fraud activities by developing a text convolutional attention network (TextCAN) to extract local features with contextual semantic relations from consumer reviews.

The reviews above have identified several issues with current fraud detection methods. For instance, standard approaches are sometimes employed without considering their performance, leading to biased results. Ensemble models are more complicated and susceptible to overfitting. Ensemble models are more complicated and susceptible to overfitting. DNNs are thought of as "black boxes" since they require a lot of data to train. As a result, developing a rule-based model is critical for financial fraud detection, regardless of any dataset imbalance concerns.

3. METHOD

The training dataset D consists of N number of transactions, $T = [T_1, T_2, T_3, \dots, T_N]$ and each transaction is characterized by attributes $A = [A_1, A_2, A_3, \dots, A_m]$. The Limit is set for each attribute in D dataset. For example, if we have feature A , the limit L will be set from A values to apply conditions like if the value is less than L or more significant than L , the class value will be either 0 or 1, whereas 0 and 1 represent the class of a transaction as non-fraud and fraud respectively. The following steps describe the underlying idea behind extracting the relational rules from the imbalanced financial dataset to detect fraudulent transactions. Figure 1 provides a flowchart of the suggested rule-based method. Algorithm 1 shows the complete financial fraud detection system procedure using the proposed rule-based model.

3.1. Feature selection

A huge dataset can sometimes be difficult to manage, which may lead to poor efficiency, so feature selection is a key step to remove extraneous data from a comprehensive dataset. In this work, we offer a method that adjusts the dynamic process by running a loop around the dataset to gather the significance of its characteristics and automatically filter out the less significant aspects. As a result, the model can only retain the crucial and applicable elements. Consequently, the model's precision and effectiveness are increased. There are several methods for choosing features, including the chi-square, Baruta, DT, and RF methods.

Using an iterative RF approach, Baruta is another automatic feature removal system. This method applies RF repeatedly while iterating the dataset. This method is therefore expensive, time-consuming, and unsuited for huge datasets. In this study, the first 80% of its key features are selected using an RF, and the remaining 20% are selected using a DT, which produces more optimal outcomes. Without the need for human input, our structure will cycle through the dataset, determine which features are most significant, and discard those that are not. Thus, this model chooses 9 features from the actual PaySim dataset's 11 available features.

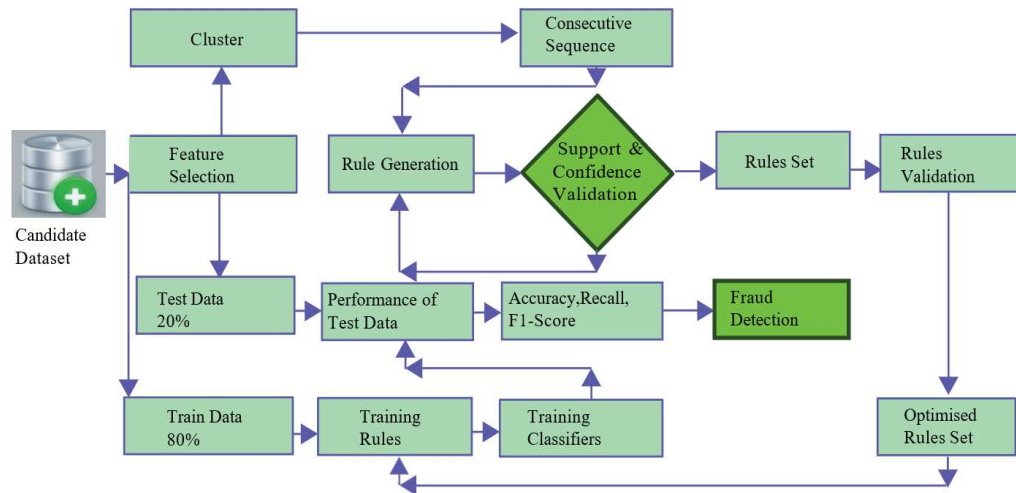


Figure 1. Flowchart of proposed rule-based model

Algorithm 1. Rule-based model algorithm

Data: Input data set (DS), a dataset containing n financial transactions

Result: Fraud and non-fraud transactions

```

if DS.dataType==numeric then
    convert data into numeric;
end
Feed DS into the RandomForest model and check importance;
if featureImportance≤minimumThreshold then
    eliminate feature;
end
Feed the DS with important features into the DecisionTree model;
BalanceDataset ← Over Sampling with AHS;
while i in transactions do
    if is Fraud==1 then
        if type==Cluster[key] then
            Cluster[key].append(i)
        end
    end
end
ConsecutiveSequence() ← Find Consecutive sequence from each cluster ;
while clusterIndex in Cluster do
    while amount in ConsecutiveSequence do
        if confidence≥ConfidenceThreshold then
            Confidence[cI][amount].append(confidence)
        end
        if support≥SupportThreshold then
            Support[cI][amount].append(support)
        end
    end
end
AllPossibleRules() ← Create all possible relational rules for each category;
while ruleIndex in AllPossibleRules do
    support() ← Calculate support of rule
    Confidence() ← Confidence of rule
    if confidence≥confidenceThreshold && support ≥ supportThreshold then
        RuleSet.append(rule)
    end
end
end
  
```

```
prediction ← Each transaction is validated by the rule set
return prediction;
```

3.2. Cluster

From a financial dataset, such as the PaySim dataset, which contains transaction types like CASH IN, CREDIT, CASH OUT, TRANSFER, DEBIT, and PAYMENT, fraud detection association rules can be generated, we can use a clustering algorithm to identify patterns and group similar transactions together. One commonly used algorithm for clustering is the k-means algorithm. In this case, we want to cluster the dataset based on the transaction types where fraud occurs, namely CASH IN, CASH OUT, TRANSFER, DEBIT, and PAYMENT. To apply the k-means clustering algorithm, we use the following steps: i) data preparation: convert the dataset into a suitable format for clustering. Each transaction in the dataset can be represented as a vector of binary variables, indicating whether a specific transaction type is present. For example, a transaction with CASH IN, CREDIT, and TRANSFER can be represented as a vector [1, 0, 1, 0, 0, 0]; ii) initialization: depending on the anticipated number of fraud transaction patterns, choose K the number of clusters. Randomly initialize K cluster centroids. These centroids will represent the transaction patterns associated with the fraud; iii) assignment: calculate the distance between each transaction vector and the cluster centroids. Assign each transaction to the cluster with the nearest centroid based on a distance metric such as Euclidean distance. The distance can be computed using (1) [1]:

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Where (x_1, y_1) and (x_2, y_2) are the coordinates of the two points being compared (transaction vectors and cluster centroids); iv) update: after assigning all transactions to clusters, update the centroids by computing each cluster's mean of the transaction vectors; v) repeat steps iii and iv: achieve convergence by repeating the assignment and updating stages. Once a certain number of iterations have been completed or the centroids no longer exhibit considerable variation, convergence occurs.

Once the clustering process is completed, we generate clusters representing different fraud transaction patterns. We can then analyze these clusters to generate fraud detection association rules. Association rules can provide insights into each cluster's relationships between different transaction types. For example, we may observe that TRANSFER transactions often follow CASH OUT transactions within a specific cluster. This association rule could indicate a potential fraudulent behavior pattern. Using clustering algorithms like k-means, we can identify and group transactions with similar characteristics, allowing us to detect potential fraud patterns and generate applicable association rules for fraud detection. The financial transaction datasets contain a variety of transaction types. Certain types of transactions are susceptible to fraud. To analyze these occurrences, the dataset is segmented into distinct groups based on transaction types, including CASH IN, CASH OUT, TRANSFER, DEBIT, and PAYMENT, where instances of fraudulent transactions occur. As an illustration, in the case of the PaySim dataset, which comprises five types of transactions-CASH IN, CASH OUT, TRANSFER, DEBIT, and PAYMENT, the dataset is partitioned into five clusters for the purpose of generating rules.

3.3. Consecutive sequence

The consecutive sequences are calculated from each type of transaction from each cluster. Algorithm 2 shows the procedure of consecutive sequences. Let's think about a set of values of amount attribute: 1000, 1001, 2000, 3000, 5000, 4000, 5001, 1003, 2001, 3001, 4001, 2003, 2004, 2004, 1002, 5003, 5004, 5005, 4001, 3002, 1004, 4003, 4002, 1004, 6000, 3003.

From the above values, the consecutive sequences are as follows: i) 1000, 1001, 1002, 1003, 1004, 1004, ii) 2000, 2001, 2003, 2004, 2004, iii) 3000, 3001, 3002, 3003, iv) 4000, 4001, 4001, 4002, 4003, v) 5000, 5001, and vi) 6000. To create relational rules that distinguish the proposed rule-based model from previous rule-based models like Apriori and FP-Growth, the maximum and minimum limits of each sequence are calculated. The minimum and maximum limit of sequences 1, 2, 3, 4, and 5 are 1000 and 1004, 2000 and 2004, 3000 and 3003, 4000 and 4003, and 5000 and 5001, respectively, whereas the minimum and maximum of sequence 6 are 6000 due to only one value. As a result, the rule terms of amount attribute are $1000 \leq \text{amount} \leq 1044 = \text{amount}$, $2000 \leq \text{amount} \leq 2004 = \text{amount}$, $3000 \leq \text{amount} \leq 3003 = \text{amount}$, $4000 \leq \text{amount} \leq 4003 = \text{amount}$, $5000 \leq \text{amount} \leq 5001 = \text{amount}$, and $\text{amount} = 6000$.

Similarly, let's consider a set of values of the oldbalanceOrg attribute: 9203, 9200, 9201, 6099, 7000, 7001, 6301, 6302, 6303, 5501, 5502, 5503, 5504, 5000, 5001, 5003, 5004, 5005, 5002, 4001, 4001, 4003, 1003, 1004, 1000, 1001, 1002, 999.

From the above values, the consecutive sequences are as follows: i) 999, 1000, 1001, 1002, 1003, 1004, ii) 4001, 4001, 4003, iii) 5000, 5001, 5002, 5003, 5004, 5005, iv) 5501, 5502, 5503, 5504, v) 6301, 6302, 6303, vi) 6099, 7000, 7001, vii) 9200, 9201, 9203. After calculating the minimum and maximum

values of each sequence, the rule terms of the oldbalanceOrg(OBO) attribute are formed as follows: 999 <=OBO && 1004>=OBO, 4001 <=OBO && 4003>=OBO, 5000 <=OBO && 5005>=OBO, 6301 <=OBO && 6303>=OBO, 6099 <=OBO && 7001>=OBO, and 9200 <=OBO && 9203>=OBO.

Following the above rule terms of amount and oldbalanceOrg attributes, the rule terms of other attributes such as newbalanceOrig, oldbalanceDest, and newbalanceDest are formed. From the experiment dataset, the rule terms of amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, and newbalanceDest are 578, 566, 10, 415, and 585, respectively. If we apply " C_r " on each attribute of the PaySim dataset to form the rules, then according to the proposed model, the possible minimum number of rules is $^{578}C_1 \times ^{566}C_1 \times ^{10}C_1 \times ^{415}C_1 \times ^{585}C_1$.

Algorithm 2. Consecutive sequence algorithm

```
Data: List, N
Result: List of consecutive sequences
if N == 0 then
    return List;
end
while index in range(1, n + 1) do
    if index == 1 or List[index] - List[index - 1] != 1 then
        if length == 1 then
            item ← a[index - length] if length == 1 then
                item ← a[indexlength]
            else
                List.append(item)
            end
        else
            temp ← a[index-length] List.append(temp) length ← 1
        end
    else
        length ← length + 1
    end
end
return List;
```

3.4. Rule generation

An association rule is represented as $A \rightarrow C$, where A is defined as the antecedent that consists of different rule terms with (and) relations and C as the consequent whereas each has support (SUP) and confidence (CON). The proposed rule-based model generates relational rules with the consequent containing only fraud. We use an unsupervised process for rule generation for the fraud class. The ruleset is initially configured as an empty set $R = \emptyset$, and as time goes on, new rules R_i are generated and added to this set based on how well they perform on the dataset for the fraud class under consideration. During the rule learning process for the fraud class, each straightforward rule is applied to the dataset and either added to the ruleset or dismissed. Multiple rules may occasionally be combined or separated to optimize performance. In this step, the relational rules are developed by combining one rule term of each attribute with their transaction type. Following support, the Apriori and frequent pattern growth (FP-Growth) generate rules based on combinations of transactional elements. All payment transaction datasets are numerical, hence Apriori and FP-Growth are inappropriate for them. Tables 1 and 2 show some generated rules by the proposed rule-based model using PaySim and BankSim datasets respectively.

Table 1. Some generated relational association rules using BankSim dataset

Relational association rule	SUP	CON
{gen==F, type=="es health", amount>78.96, 3.93<=amount, age>=25}→{Fraud}	17%	53%
{gen==M, type=="es health", amount<78.49, 3.63<=amount, age>=26}→{Fraud}	65%	29%
{gen==F, type=="es wellnessandbeauty", amount<79.22, 3.27<=amount, age>=32}→{Fraud}	94%	59%
{gen==M, type=="es wellnessandbeauty", amount<76.94, 4.95<=amount, age>=34}→{Fraud}	25%	15%
{gen==F, type=="es barsandrestaurants", amount<78.19, 5.11<=amount, age>=28}→{Fraud}	19%	52%

Table 2. Some generated relational association rules using PaySim dataset

Relational association rule	SUP	CON
{oldbalanceOrg==amount, type==CASH OUT}→{Fraud}	72%	98%
{oldbalanceOrg<=56900, type==CASH OUT, newbalanceDest<=105}→{Fraud}	49%	13%
{oldbalanceDest==newbalanceDest=0, oldbalanceOrg>0, type==CASH OUT}→{Fraud}	64%	82%
{oldbalanceOrg==amount, type==TRANSFER}→{Fraud}	45%	12%
{oldbalanceOrg<=56900, type==TRANSFER, newbalanceDest<=105}→{Fraud}	20%	47%

3.5. Support and confidence validation

Support and confidence are two measures that are commonly used to evaluate the strength of association rules in machine learning. Minimum support is the minimum frequency at which a rule must occur in the dataset to be considered significant. A rule that has low support may be considered irrelevant or spurious. When the prerequisites of the rule are met, confidence is a measure of how frequently a rule is correct and is determined as the ratio of the number of times the rule is correct to the total number of times it is applicable. A rule with low confidence may be unreliable and need to be refined. The support and confidence of each rule are compared with the user-defined threshold. After iterating through each sequence of clusters and getting the sequence of clusters containing the frequency of each condition being satisfied, all the items of these sequences of clusters are passed to check the threshold function. In this function, the support and confidence value of each sequence of cluster items is calculated and compared with the support and confidence threshold set by the user.

3.6. Rules set

Refining the rule set can be done by selecting the most relevant rules based on the minimum support and confidence concepts. However, not all rules are equally important, and some may be misleading. To ensure the rule set is efficient and effective, the model evaluates each rule based on its support and confidence. The minimum support is the minimum number of times a rule occurs in the dataset, while the confidence measures how often the rule is correct. The selection process of the rules based on minimum support and confidence ensures that the rules capture meaningful patterns in the data and avoid unreliable or spurious rules. The user-defined minimal support and confidence levels are then used to create the refined rule set, which is then established by selecting only those rules that do so. As a result, a rule-based machine learning model with improved accuracy and dependability may more precisely identify fraudulent financial transactions.

3.7. Rules validation

The rule validation step is crucial in ensuring the accuracy and effectiveness of the rules. The rule validation is performed through the following methods: i) Rule structure verification: For example, if a rule is generated based on a dataset of customer transactions, such as “IF the transaction is a CASH-OUT and the amount is greater than \$1000 THEN flag it as potential fraud”, the rule structure verification checks whether this rule follows the IF-THEN structure. And ii) Rule consistency verification: For instance, consider the following rule generated from the same dataset, “IF the transaction is a CASH IN and the amount is less than \$500 THEN the transaction is not considered fraudulent”. The rule consistency verification ensures that this rule is consistent with other association rules in the repository, particularly in terms of the antecedent and consequent constraints. This is important because conflicting rules may lead to inaccurate predictions, and the reasoner is used to identify and remove any inconsistent rules from the association rule repository.

3.8. Optimized rules set

Rule optimization is the process of eliminating any rules that do not enhance the classifier's performance. On the dataset, we iteratively explored to determine the importance of support, confidence, and redundancy for each rule. A ruleset with more redundant rules has lower support and confidence thresholds. Conversely, using a confidence threshold value of 50% to 100% results in the maximum number of positively anticipated fraud data transactions and the least amount of rule redundancy. As a result, we decided that the confidence criterion should be confidence $\geq 50\%$. Also, we looked at whether a simple rule can combine with other rules to achieve the best fitness value. If, after combining with other rules, a simple rule with below threshold fitness may provide the highest fitness, we consider that simple rule to be significant. When r_1 and r_2 are combined, we receive the highest fitness even though r_1 's confidence is lower than the confidence threshold. Hence, as a last step, we consider $[(r_1 \ r_2) \ r_3] \rightarrow \text{Fraud}$ and prune the other rules during the optimization process. Figure 2 shows the rules optimization process.

3.9. Fraud detection

The proposed rule-based model generates a set of rules that are used in a prediction function for fraud detection, where they are converted into IF-ELSE statements. For instance, the generated rule “IF the transaction is a TRANSFER and the amount is greater than \$10,000 THEN flag it as potential fraud” can be converted into the IF-ELSE statement: “IF transaction type is TRANSFER AND amount \$10,000 THEN flag as fraud ELSE continue processing”. These rules are then applied to new transactions in real-time to detect any fraudulent activities. If a transaction violates one or more of the fraud rules, it is flagged as suspicious and may be subjected to further investigation. For example, if a new transaction is a TRANSFER of \$15,000, it will be flagged as potential fraud as it violates the fraud rule mentioned above.

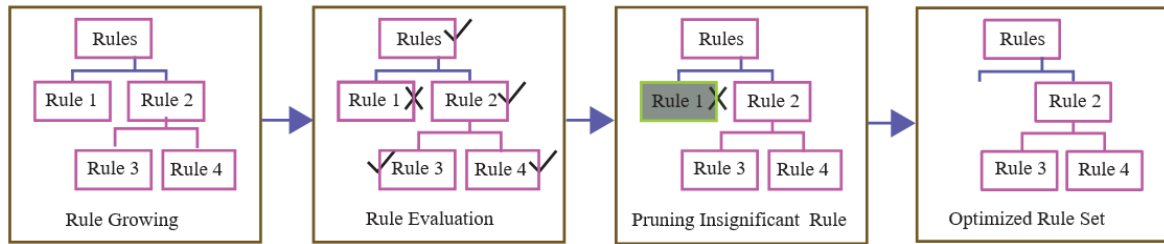


Figure 2. Rule optimization process

4. RESULT AND DISCUSSION

4.1. Dataset

Using two unbalanced datasets from the machine learning website www.kaggle.com, we evaluated the performance of our suggested model. They are PaySim dataset [28] and BankSim dataset [29]. The PaySim dataset consists of 6,362,620 card transactions, out of which 6,354,407 are valid and 8,213 are fraudulent. The dataset has the following 11 attributes: step, type, amount, oldbalanceOrg, newbalanceOrg, nameOrig, oldbalanceDest, newbalanceDest, isFraud, and isFlaggedFraud. In the BankSim dataset, there are 10 different attributes: step, customer, age, gender, zipcodeOri, merchant, zipMerchant, category, amount, and fraud. There are 594,643 records in all, including 7,200 fraudulent transactions and 587,443 valid payments in the dataset.

4.2. Evaluation method

We experimented with the original datasets to compare the performance of the proposed rule-based model with that of various classifiers, including RF, DT, MLP, KNN, NB, and LR. The Python programming language and its machine-learning modules were used to carry out the tests. The dataset was divided into training and test sets, with 80% of the samples being used for training and 20% being utilized to test the rule-based model's performance outcomes. Using metrics like accuracy, precision, recall, F1-score, specificity, confusion matrix, MCC, and AUC, the performance of machine learning classification algorithms is assessed after they have been trained on the dataset. The proportion of accurately predicted labels among all labels is what is known as accuracy. The percentage of accurately predicted fraudulent samples by the classifier is known as recall, also known as sensitivity. Specificity, also known as the true negative rate, on the other hand, refers to the proportion of valid transactions that were precisely predicted. In a binary labeled dataset, precision is the proportion of positively predicted labels that were correctly made out of all the positive labels. F1-Score returns the weighted average of recall and precision. There is no ideal metric for evaluating the effectiveness of a model. Since an AUC value of 1 indicates a perfect model, the closer a classifier's AUC value is to 1, the better. The ROC curve compares the ratio of true positives to false positives at different threshold levels. Data regarding a classifier's expected and actual classifications, such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), are included in a confusion matrix [30]. The MCC is the greatest overall metric, even if there is not a perfect way to tell the difference between true and false positives and negatives based on just one indicator. A flawless prediction is indicated by an MCC result of +1, whereas a total disagreement is indicated by a value of 1. MCC can be calculated using (2):

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (2)$$

4.3. Result analysis

The data from the tests conducted for this research were used to train the suggested rule-based model as well as the other classifiers. The outcomes are displayed in Tables 3 and 4. Firstly, experimental results for the proposed method achieved MCC, ROC-AUC, accuracy, precision, recall, and F1-score of 0.993, 0.991, 0.996, 0.987, and 0.998 for PaySim dataset, and 0.995, 0.973, 0.998, 0.997, 0.987, and 0.989 for BankSim dataset. The experiment enhanced the proposed rule-based model's performance more than the other classifiers already in use. The highest MCC score is 0.993 and 0.995 for PaySim and BankSim datasets, which indicates the proposed rule-based model's better and more robust performance. The enhanced precision values are significant since precision is a key statistic in fraud detection. Figures 3(a) and 3(b) demonstrate that the ROC curve of the proposed rule-based model is closer to the upper-left corner, indicating stronger predictiveness compared to other classifiers, while the ROC curve that is used to explain the trade-off between a true-positive rate and a false-positive rate is used to highlight the trade-off between a true-positive rate and a false-positive rate. Additionally, the proposed model outperformed with an AUC

value of 0.991 for the PaySim dataset and 0.973 for the BankSim dataset. According to these results, the proposed model performed well in identifying fraudulent and legal transactions. The proposed model's performance on the PaySim and BankSim datasets can be evaluated by analyzing the results presented in Figures 4(a) and 4(b). In Figure 4(a), the proposed model achieved a high number of correct predictions, with a true positive (TP) rate of 98.18% and a true negative (TN) rate of 0.10%. The model's incorrect predictions consisted of a false positive (FP) rate of 1.71% and a false negative (FN) rate of 0%. Similarly, in Figure 5, the proposed model's performance on the BankSim dataset showed a TP rate of 97.22% and a TN rate of 0.09%. The model's incorrect predictions included an FP rate of 2.61% and an FN rate of 0.08%. It is worth noting that the TP rate remained high in both datasets, indicating that the proposed model is effective at identifying positive instances. Overall, the results suggest that the proposed model is capable of making accurate predictions on both the PaySim and BankSim datasets, with a relatively low rate of false positives and false negatives. Figures 5(a) and 5(b) compare the precision values of different models with the proposed rule-based model, while Figures 6(a) and 6(b) compare the specificity values of different models with the proposed rule-based model. Figures show that the suggested rule-based model performed substantially better than other classifiers. High specificity means the model is correctly detecting negative cases, whereas high precision means the model is correctly identifying positive ones. The proposed rule-based model greatly increased the specificity that is the highest among the other classifiers using the PaySim and BankSim datasets respectively. The improved performance of the suggested strategy is shown in Figures 7(a)-7(b), as it generates fewer rules during the experiment compared to Apriori and FP growth algorithms for all candidate datasets. Using PaySim and BankSim datasets, our method generates 1,264 and 1,250 association rules, respectively.

Table 3. Experiment results using PaySim dataset

Method	MCC	ROC-AUC	Accuracy	Precision	Recall	F1-Score
RF	0.985	0.888	0.946	0.976	0.947	0.946
DT	0.832	0.919	0.936	0.967	0.938	0.938
MLP	0.723	0.694	0.493	0.975	0.497	0.478
KNN	0.956	0.940	0.865	0.946	0.867	0.868
NB	0.658	0.619	0.715	0.964	0.717	0.696
LR	0.936	0.957	0.934	0.975	0.936	0.935
RBM	0.993	0.991	0.996	0.998	0.987	0.998

Table 4. Experiment results using BankSim dataset

Method	MCC	ROC-AUC	Accuracy	Precision	Recall	F1-Score
RF	0.968	0.967	0.964	0.945	0.978	0.975
DT	0.978	0.952	0.967	0.966	0.987	0.988
MLP	0.927	0.952	0.916	0.977	0.978	0.947
KNN	0.935	0.967	0.925	0.976	0.987	0.995
NB	0.956	0.963	0.945	0.926	0.927	0.968
LR	0.963	0.952	0.953	0.994	0.947	0.978
RBM	0.995	0.973	0.998	0.997	0.987	0.989

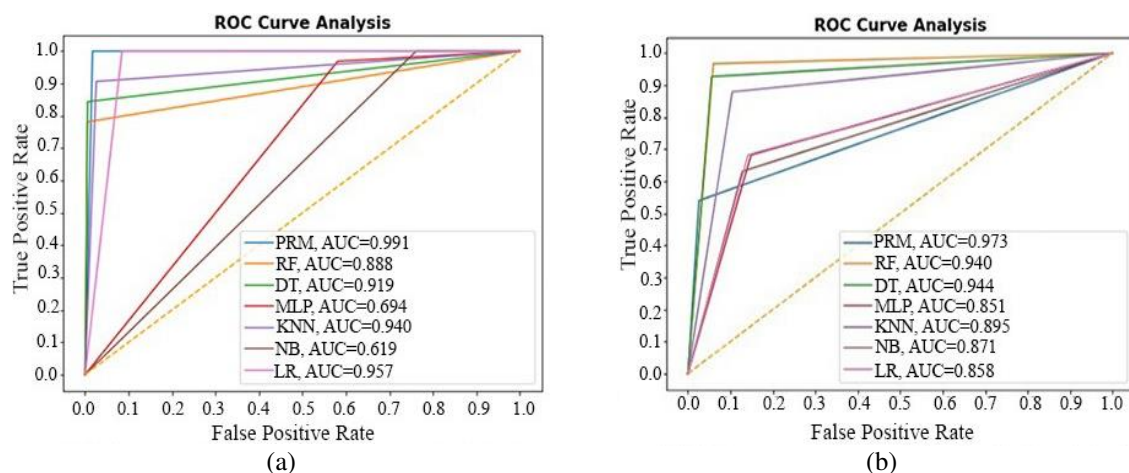


Figure 3. ROC curve of the various model (a) using PaySim dataset and (b) using BankSim dataset

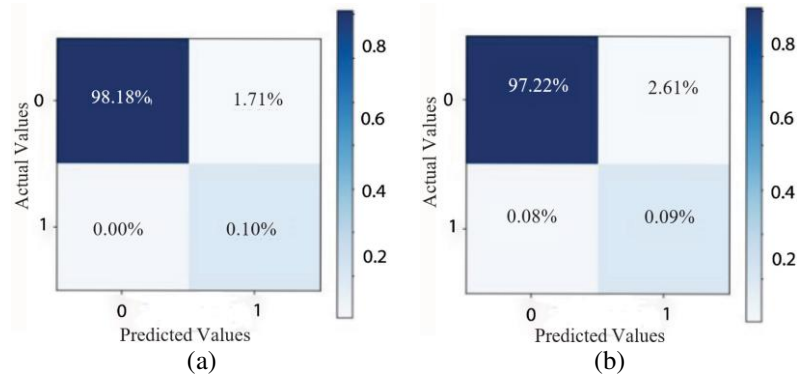


Figure 4. Confusion matrix of the proposed rule-based model (a) with PaySim dataset and (b) with BankSim dataset

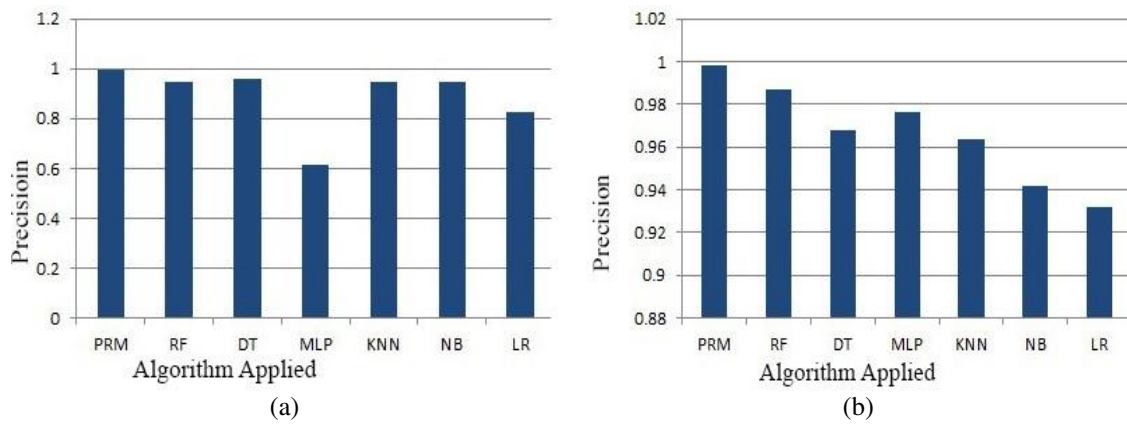


Figure 5. The precision of various classifiers (a) using PaySim dataset and (b) using BankSim dataset

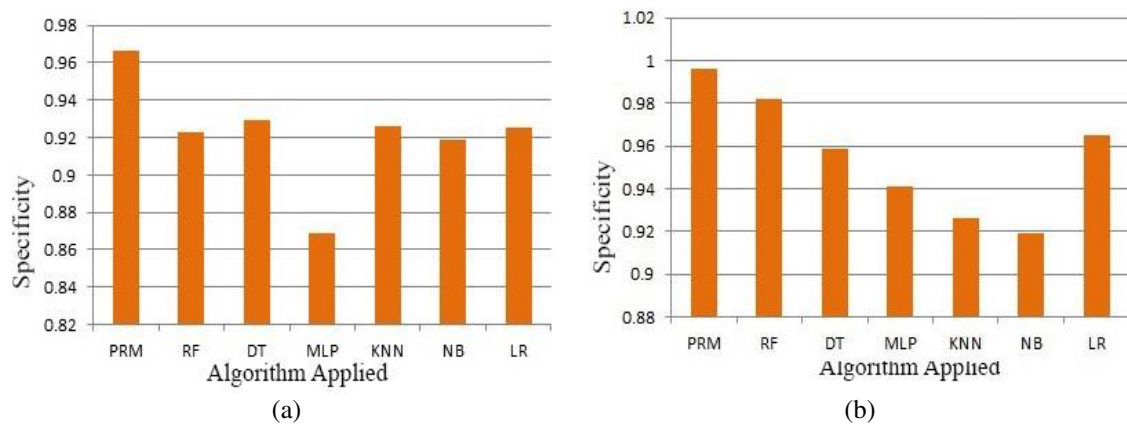


Figure 6. Specificity of various classifiers (a) using PaySim dataset and (b) using BankSim dataset

In contrast, the Apriori algorithm produces 12,800 and 13,060 association rules for PaySim and BankSim datasets, respectively, while the FP growth algorithm produces 11,356 and 11,096 association rules for PaySim and BankSim datasets, respectively. The results indicate that traditional algorithms consider all possible combinations of attributes, resulting in a large number of association rules, while our approach generates fewer rules. Our method outperforms than the conventional association rule mining algorithms as it discards redundant rules and retains non-redundant ones, resulting in a smaller but more effective set of association rules.

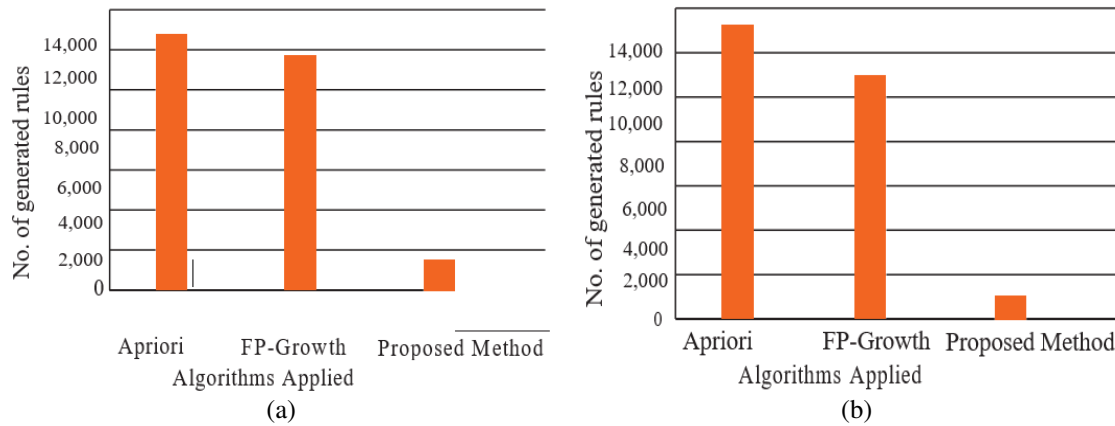


Figure 7. Generated rules comparison (a) using PaySim dataset and (b) using BankSim dataset

4.4. Comparison with existing methods

Comparing our proposed technique to traditional algorithms does not demonstrate its superior performance. To contrast our strategy with other financial fraud detection strategies already being used in the literature. The techniques include the sequential combination of a C4.5 DT and NB [31], a LightGBM with a Bayesian-based hyperparameter optimization algorithm [28], a cost-sensitive SVM (CS SVM) [32], an optimized RF classifier [33], a random forest classifier with SMOTE data resampling [34], an improved AdaBoost classifier with PCA and SMOTE method [35], a cost-sensitive neural network ensemble (CS-NNE) [36], and a model based on overfitting-cautious heterogeneous ensemble (OCHE) [37]. In Table 5, the proposed rule-based model demonstrates excellent performance compared to the other cutting-edge approaches, demonstrating the robustness of the suggested method.

Table 5. Comparative results with existing models

Reference	Method	Sensitivity	Specificity	AUC
Kalid <i>et al.</i> [31]	C4.5+NB	0.872	1	-
Taha and Malebary [17]	LightGBM	-	-	0.928
Makki <i>et al.</i> [32]	CS SVM	0.650	-	0.620
Khatri <i>et al.</i> [33]	Optimized random forest	0.782	-	-
Mrozek <i>et al.</i> [35]	Random forest+SMOTE	0.829	-	0.910
Zhou <i>et al.</i> [36]	AdaBoost+SMOTE+PCA	-	-	0.965
Yotsawat <i>et al.</i> [37]	CS-NNE	-	0.936	0.980
This paper with PaySim dataset	Rule-Based Model+AHS	0.999	0.998	0.997
This paper with BankSim dataset	Rule-Based Model+AHS	0.998	0.978	0.973

5. CONCLUSION

Financial fraud is a significant problem that impacts both private citizens and business entities, costing the economy billions of dollars annually. In order to avoid the use of resampling, this study suggests a rule-based fraud detection approach that has proven to be quite successful at identifying financial fraud. The experimental outcomes show that the suggested method performs better than the current methods, obtaining a detection level of 98% out of 1. The proposed rule-based model has demonstrated robustness by achieving the highest MCC score of 99% on both datasets. The proposed rule-based model offers transparency and interpretability in the learning process, which is crucial for the financial sector. This research highlights the potential benefits of using rule-based models with novel resampling techniques for financial fraud detection in machine learning. Therefore, the proposed method can serve as an efficient tool for detecting fraud in financial transactions on both balanced and imbalanced datasets. In future work, we explore newer techniques to reduce the rule generation and classification process time, leading to further improvements in financial fraud detection. This will help identify and prevent fraudulent transactions in the future, which will reduce the amount of losses faced in the financial sector every day.

REFERENCES




- [1] "Oxford learner's dictionaries," *Oxford University Press*. <https://www.oxfordlearnersdictionaries.com/definition/english/fraud> (accessed Oct. 26, 2021).

- [2] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card Fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, pp. 14277–14284, 2018, doi: 10.1109/ACCESS.2018.2806420.
- [3] M. Zareapoor and J. Yang, "A novel strategy for mining highly imbalanced data in credit card transactions," *Intelligent Automation and Soft Computing*, pp. 1–7, May 2017, doi: 10.1080/10798587.2017.1321228.
- [4] M. Óskarsdóttir, C. Bravo, C. Sarraute, J. Vanthienen, and B. Baesens, "The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics," *Applied Soft Computing*, vol. 74, pp. 26–39, Jan. 2019, doi: 10.1016/j.asoc.2018.10.004.
- [5] B. Bandaranayake, "Fraud and corruption control at education system level," *Journal of Cases in Educational Leadership*, vol. 17, no. 4, pp. 34–53, Dec. 2014, doi: 10.1177/1555458914549669.
- [6] C. Liu *et al.*, "Fraud transactions detection via behavior tree with local intention calibration," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2020, pp. 3035–3043, doi: 10.1145/3394486.3403354.
- [7] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, Oct. 2020, pp. 315–324, doi: 10.1145/3340531.3411903.
- [8] J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," *Computers and Security*, vol. 57, pp. 47–66, Mar. 2016, doi: 10.1016/j.cose.2015.09.005.
- [9] W. Zhou and G. Kapoor, "Detecting evolutionary financial statement fraud," *Decision Support Systems*, vol. 50, no. 3, pp. 570–575, Feb. 2011, doi: 10.1016/j.dss.2010.08.007.
- [10] H. Tingfei, C. Guangquan, and H. Kuihua, "Using variational auto encoding in credit card Fraud detection," *IEEE Access*, vol. 8, pp. 149841–149853, 2020, doi: 10.1109/ACCESS.2020.3015600.
- [11] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, Feb. 2011, doi: 10.1016/j.dss.2010.08.008.
- [12] D. Sánchez, M. A. Vila, L. Cerda, and J. M. Serrano, "Association rules applied to credit card fraud detection," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3630–3640, Mar. 2009, doi: 10.1016/j.eswa.2008.02.001.
- [13] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A neural network ensemble with feature engineering for improved credit card Fraud detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022, doi: 10.1109/ACCESS.2022.3148298.
- [14] N. Nguyen *et al.*, "A proposed model for card Fraud detection based on CatBoost and deep neural network," *IEEE Access*, vol. 10, pp. 96852–96861, 2022, doi: 10.1109/ACCESS.2022.3205416.
- [15] S. K. Hashemi, S. L. Mirtaheri, and S. Greco, "Fraud detection in banking data by machine learning techniques," *IEEE Access*, vol. 11, pp. 3034–3043, 2023, doi: 10.1109/ACCESS.2022.3232287.
- [16] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card Fraud detection using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021, doi: 10.1109/ACCESS.2021.3134330.
- [17] A. A. Taha and S. J. Malebary, "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020, doi: 10.1109/ACCESS.2020.2971354.
- [18] D. Jiaman, Z. Shujie, L. Runxin, F. Xiaodong, and J. Lianyin, "Association rules-based classifier chains method," *IEEE Access*, vol. 10, pp. 18210–18221, 2022, doi: 10.1109/ACCESS.2022.3149012.
- [19] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *Journal of Big Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00573-8.
- [20] S. Bakhtiari, Z. Nasiri, and J. Vahidi, "Credit card fraud detection using ensemble data mining methods," *Multimedia Tools and Applications*, vol. 82, no. 19, pp. 29057–29075, Aug. 2023, doi: 10.1007/s11042-023-14698-2.
- [21] R. Van Belle, B. Baesens, and J. De Weerd, "CATCHM: A novel network-based credit card fraud detection method using node representation learning," *Decision Support Systems*, vol. 164, Jan. 2023, doi: 10.1016/j.dss.2022.113866.
- [22] Z. Salekshahrezaee, J. L. Leevy, and T. M. Khoshgoftaar, "The effect of feature extraction and data sampling on credit card fraud detection," *Journal of Big Data*, vol. 10, no. 1, Jan. 2023, doi: 10.1186/s40537-023-00684-w.
- [23] H. Fanai and H. Abbasimehr, "A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection," *Expert Systems with Applications*, vol. 217, May 2023, doi: 10.1016/j.eswa.2023.119562.
- [24] H. Ahmad, B. Kasasbeh, B. Aldabaybah, and E. Rawashdeh, "Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS)," *International Journal of Information Technology*, vol. 15, no. 1, pp. 325–333, Jan. 2023, doi: 10.1007/s41870-022-00987-w.
- [25] L. Ni, J. Li, H. Xu, X. Wang, and J. Zhang, "Fraud feature boosting mechanism and spiral oversampling balancing technique for credit card fraud detection," *IEEE Transactions on Computational Social Systems*, pp. 1–16, 2023, doi: 10.1109/TCSS.2023.3242149.
- [26] B. Karunachandra, N. Putera, S. R. Wijaya, D. Suryani, J. Wesley, and Y. Purnama, "On the benefits of machine learning classification in cashback fraud detection," *Procedia Computer Science*, vol. 216, pp. 364–369, 2023, doi: 10.1016/j.procs.2022.12.147.
- [27] S. Lai, J. Wu, Z. Ma, and C. Ye, "BTextCAN: Consumer fraud detection via group perception," *Information Processing and Management*, vol. 60, no. 3, May 2023, doi: 10.1016/j.ipm.2023.103307.
- [28] E. Lopez-Rojas, "Synthetic financial datasets for fraud detection," *Kaggle*. <https://www.kaggle.com/datasets/ealaxi/paysim1> (accessed Dec. 20, 2022).
- [29] E. Lopez-Rojas, "Synthetic data from a financial payment system," *Kaggle*. <https://www.kaggle.com/datasets/ealaxi/banksim1> (accessed Dec. 20, 2022).
- [30] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd Edition. Elsevier, 2011.
- [31] S. N. Kalid, K.-H. Ng, G.-K. Tong, and K.-C. Khor, "A multiple classifiers system for anomaly detection in credit card data with unbalanced and overlapped classes," *IEEE Access*, vol. 8, pp. 28210–28221, 2020, doi: 10.1109/ACCESS.2020.2972009.
- [32] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019, doi: 10.1109/ACCESS.2019.2927266.
- [33] S. Khatri, A. Arora, and A. P. Agrawal, "Supervised machine learning algorithms for credit card fraud detection: a comparison," in *2020 10th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, Jan. 2020, pp. 680–683, doi: 10.1109/Confluence47617.2020.9057851.
- [34] K. I. Alkhatib, A. I. Al-Aiad, M. H. Almahmoud, and O. N. Elayan, "Credit card fraud detection based on deep neural network approach," in *2021 12th International Conference on Information and Communication Systems (ICICS)*, May 2021, pp. 153–156, doi: 10.1109/ICICS52457.2021.9464555.




- [35] P. Mrozek, J. Panneerselvam, and O. Bagdasar, "Efficient resampling for fraud detection during anonymised credit card transactions with unbalanced datasets," in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, Dec. 2020, pp. 426–433, doi: 10.1109/UCC48980.2020.00067.
- [36] H. Zhou, L. Wei, G. Chen, P. Lin, and Y. Lin, "Credit card fraud identification based on principal component analysis and improved Adaboost algorithm," in *2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, Dec. 2019, pp. 507–510, doi: 10.1109/ICICAS48597.2019.00111.
- [37] W. Yotsawat, P. Wattuya, and A. Srivihok, "A novel method for credit scoring based on cost-sensitive neural network ensemble," *IEEE Access*, vol. 9, pp. 78521–78537, 2021, doi: 10.1109/ACCESS.2021.3083490.

BIOGRAPHIES OF AUTHORS






Saiful Islam    received the B.Sc. degree in computer science and engineering from Dhaka University of Engineering and Technology, Bangladesh. He is a student of M.Sc. program in the Department of Computer Science and Engineering at Chittagong University of Engineering and Technology, Bangladesh. He is a full-time lecturer with the Department of Computer Science and Engineering, International Islamic University of Chittagong, Bangladesh. His research interests include big data, artificial intelligence, IoT, NLP, cyber security, machine learning and deep learning for software industries. He can be contacted by email at engsaiful0@gmail.com.



Md. Mokammel Haque    received the B.Sc. Eng. degree in computer science and engineering from the Chittagong University of Engineering and Technology (CUET), Bangladesh, the M.Sc. degree in computer engineering from Kyung Hee University, South Korea, and the Ph.D. degree in computing from Macquarie University, Australia. He is currently working as a full-time professor with the Department of Computer Science and Engineering, CUET. His research interests include cybersecurity, computer networks, cryptography, machine learning, and algorithms. He can be contacted by email at mokammel@cuet.ac.bd.



Abu Naser Mohammad Rezaul Karim    obtained B.Sc. (honors) and M.Sc. degree in Mathematics from University of Chittagong (CU), Bangladesh, PGD degree in ICT (Information and communication technology) from IICT, BUET and obtained Ph.D. degree from Islamic University, Kushtia, Bangladesh. Currently, he is working as a full-time professor in the Department of Computer Science and Engineering, International Islamic University Chittagong, Bangladesh. His research interests include modeling and simulation, mathematical analysis for computer science, function approximation, and optimization. He can be contacted by email at zakianaser@yahoo.com.