Journal of Computing and Communication Vol.3 , No.1 , PP. 116-131 , 2024

Fraud_Detection_ML: Machine Learning Based on Online Payment Fraud Detection

Maged Farouk [a], Nashwa S Ragab[a], Diaa Salama*[b,c], Omnia Elrashidy [a], Nada Ghorab [a], Jevana Hany [a], Alaa Amr [a], Omar Adel [a], Kriols Saad [a], Khaled Ali [a], Reda Elazab[a]

[a] Department of Business Information Systems, Faculty of Business, Alamein University, Alamein, Egypt

[b] Faculty of Computers Science, Misr International University, Cairo, Egypt

[c], Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

*Corresponding Author: Diaa Salama [***diaa.salama@miuegypt.edu.eg***]

ARTICLE DATA

Article history:
Received 08 Jan 2024
Revised 14 Jab 2024
Accepted 04 Feb 2024
Available online

Keywords:
Online payment fraud,
Machine-Learning,
gradient boosting,
CN2Rule Induction,
fraud deduction

ABSTRACT

Online payment fraud detection is crucial for safeguarding e-commerce transactions against sophisticated fraudsters who exploit system vulnerabilities. This paper proposes an efficient framework for predicting online payment fraud, employing six diverse machine learning algorithms, namely constant, CN7Rule induction, KNN, Tree, Random Forest, Gradient boosting, SVM, Logistic regression, Naive Bayes, Ada boost, Neural network, and stochastic gradient descent, on three distinct datasets. The gradient-boosting algorithm consistently outperformed others through rigorous testing, achieving an impressive accuracy rate of 99.7%. This algorithm demonstrated resilience across various testing scenarios, establishing itself as the most effective online payment fraud detection solution. With the highest accuracy score of 99.7% in all testing phases, gradient boosting is optimal for preemptive measures against fraudulent activities in electronic transactions, providing a robust defense mechanism for e-commerce platforms.

1. Introduction

Online payment fraud detection is a process that prevents fraudulent activities in online transactions. It involves device fingerprinting, geolocation, behavioral analysis, transaction tracking, and two-factor authentication. Machine learning and AI algorithms continuously adapt to new fraud strategies, and cooperation between payment service providers and financial institutions is beneficial [1].

Online payment fraud is a problem that arises from dishonest and illegal actions taken during electronic transactions. Unauthorized transactions, identity theft, compromised payment credentials, phishing, social engineering, insufficient security protocols, account takeover, difficulties with cross-border transactions, and risks associated with developing technologies are some of the major problems [2].

Machine learning is an evolving branch of computational algorithms designed to emulate human intelligence by learning from the surrounding environment. They are considered the working horse in the new era of big data. Techniques based on machine learning have been applied successfully in diverse fields ranging from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, and computational biology to biomedical and medical applications [3].

Machine learning plays a crucial role in addressing the challenge of online payment fraud by enabling automated, data-driven fraud detection and prevention systems. Here's how machine learning is applied to combat online payment fraud [4].

The main contribution of this paper follows: we use six algorithms, and we made predictions for online payment fraud; we use cross-validation (10), training (80%), and testing (20), and the best algorithm was gradient boosting with Accuracy (0.997).

The rest of the paper can be organized as follows: Machine learning is an effective technique when it comes to identifying and stopping online payment fraud. It can analyze vast data volumes, spot trends, and generate precise forecasts. Machine learning models can recognize suspicious trends and flag them by utilizing features like transaction amounts, locations, timestamps, user behavior, and device information. Human and tantalite ending Caprice is vital. The validity and applicability of training data were the focus of the Machine Emin Motel lawsuits.

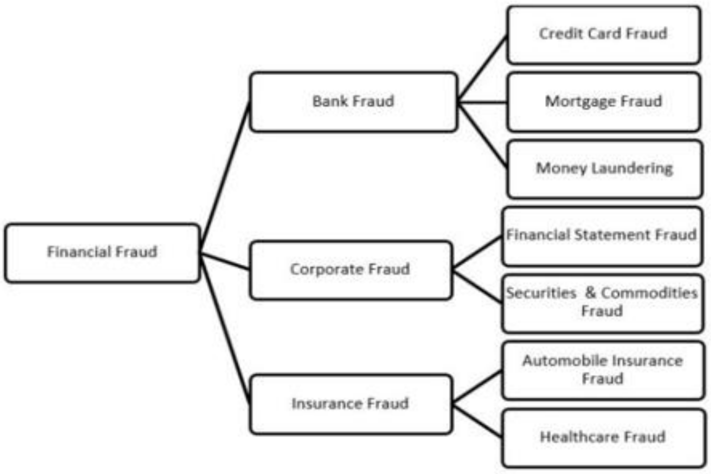

Fig1.Types of Common Fraud

2. Related Work

In [5], the authors explained that the model resulted in a considerable reduction in fraud and savings of 101,970.52 EGP out of 131,297.83 EGP. It was constructed using the IBM SPSS modeler's decision tree. It obtained an impressive 88.45% accuracy and 93.5% precision. Plotting to increase from an anticipated $10.7 billion in 2015 to $25.6 billion by the end of the decade, online and mobile fraud will significantly influence the worldwide e-payments business.

In [6], the authors explained that the model is tested against the Random Forest and Gradient Boosting Machine algorithms to determine its efficacy. Findings demonstrate the Light Gradient Boosting Machine's strong performance; in real datasets, it achieved a total recall rate of 99% and offered prompt feedback. This demonstrates how well the model detects credit card fraud.

In [7], the authors explained the process of detecting payment fraud. For this purpose, machine learning classifiers such as Bagging Ensemble Learner, C4.5 decision trees, and Naïve Bayes are suggested. These classifiers' performance is measured using evaluation measures such as Accuracy, recall rate, and precision-recall curve area rate. Three thousand two hundred ninety-three fraudulent transactions were included in the dataset, which included approximately 297,000 credit card transactions from September 2013 to November 2017. Outstanding performance is shown by machine learning classifiers, which have a precision-recall curve ratio between 99.9% and 100%. With an astounding 94.12% accuracy rate in predicting fraudulent transactions, C4.5 decision trees are the most successful classifier.

In [8] the authors explained for these detection methods, assessment criteria include specificity, Accuracy, sensitivity, and precision. Accuracy rates for Naive Bayes, K-Nearest Neighbor, Support Vector Machine, and logistic regression are 97.53%, 97.53%, 94.98%, and 99.51%, respectively. The comparative results of the study show that logistic regression is the best algorithm out of these. Unlike Naive Bayes, K-Nearest Neighbor, and Support Vector Machine, logistic regression exhibits optimal Accuracy. These results highlight the superiority of logistic regression over alternative methods in identifying credit card fraud.

In [9], the authors explained the study investigates Fraud Detection Systems (FDS) for credit cards using naïve Bayes, support vector machines, random forests, decision trees, OneR, and AdaBoost machine learning approaches. A dataset is evaluated using a variety of machine learning approaches, with an emphasis on Accuracy, to produce performance measures. The study concludes that the random forest classifier performs better than all the other techniques examined.

In [10] the authors explained the primary goal of this research is to study machine learning methods. The Ada boost algorithm and the Random Forest algorithm are the algorithms that are employed. Outcomes from both Accuracy, precision, recall, and F1-score serve as the foundation for algorithms. The confusion matrix serves as the basis for plotting the ROC curve. When comparing the Random Forest and Ada boost algorithms, the method with the highest Accuracy, precision, recall, and F1 score is deemed the most effective for fraud detection.

3. Methodology

3.1Datasets Descriptions

The first dataset consists of 10 features, and it has 1,048,576 records. The dataset was split into two partitions: 80% for training and 20% for testing. Below is a comprehensive description of each feature.
Step: An interval of time equal to one hour. Type: Indicates the kind or classification of the virtual transaction. Amount: Indicates how much money was exchanged in this transaction. NameOrig: Indicates which client started the transaction. OldbalanceOrg: This shows the customer's balance before the transaction. NewbalanceOrig: Shows the customer's balance following the transaction. NameDest: Indicates who will receive the transaction. OldbalanceDest: Stores the recipient's starting balance before the transaction. NewbalanceDest: These variables record the recipient's new balance after the transaction. IsFraud: This indicates if the transaction is thought to be fraudulent or not.

TABLE I
FEATURES OF DATASET

Features	Type	Value
step	Numerical	From 1 to 743
Type	Classification	Payment or transfer or debit. etc
amount	Numerical	0 to 92.4 m
nameOrig	Alphanumerical	String
oldbalanceOrg	Numerical	0 to 59.6m
newbalanceOrig	Numerical	0 to 49.6m
nameDest	Alphanumerical	String
oldbalanceDest	Numerical	0 to 356 m
newbalanceDest	Numerical	0 to 356 m
isFraud	Classification	0 or 1

Alphanumeric values consist of a combination of letters and numbers. This current investigation, which involved 12 undergraduates, demonstrated that angular orientation had little effect on the delay in determining whether a disoriented character was a letter or a digit [11].

3.2 Used Algorithms

These datasets were fed into twelve distinct machine learning algorithms: Gradient Boosting, K Nearest Neighbour (k-NN), and Logistic Regression. Random Forest, Decision Tree, Constant, CN7 Rule induction, SVM, Ada boost, neural network, stochastic gradient descent, and Naive Bayes algorithm. Statistics, including Accuracy, recall, precision, and MCC, were produced for each of the algorithms. Next, a chart and a comparison were made of the results. Further in the paper are the results, graphics, and a discussion.

1-Gradient Boosting: A powerful family of machine-learning algorithms known as gradient boosting machines has demonstrated notable effectiveness in various real-world applications. They can be learned in relation to various loss functions, for example, and are highly customizable to the application's specific requirements [12].
2-(k-NN): k-nearest neighbor (KNN) is one of the most prominent, simple, and basic algorithms used in machine learning and data mining. However, KNN has limited prediction ability, i.e., KNN cannot predict any instance correctly if it does not belong to any predefined classes in the training data set [13].

3-logistic regression: Logistic regression is used for binary classification based on statistical methods. It uses a linear model [14]. Hence, it is used to perform regression on a group of variables [15]. It is a normally used technique for predicting patterns in data with unambiguous or numeric attributes [14]. It uses a series of input vectors and a dependent response variable to calculate probability using a logarithm. Probability lies among the specific class. For binary classification, the response variable is given below:

$$y_i = \begin{cases} 0 \\ 1 \end{cases} \qquad (1)$$

Hence, the formula for calculating that a sample xi belongs in class one is given by

$$P(y_i = 1|x_i) = \frac{exp\, w_0 + w^T x_i}{1 + exp(w_0 + w^T x_i)} \qquad (2)$$

Where W0 and W are the regression standardization parameters, W0 represents the intercept, and W represents the coefficient vector [16].

4-Random forest: In many research contexts, random forest classification is a well-liked machine learning technique for creating prediction models. Reducing the number of variables required to produce a forecast is frequently the aim of prediction modeling, which aims to increase efficiency and lessen the workload associated with data collecting. There are several variable selection techniques available for random forest classification settings. However, there isn't much literature to advise users on which technique would be best for certain dataset types [17].

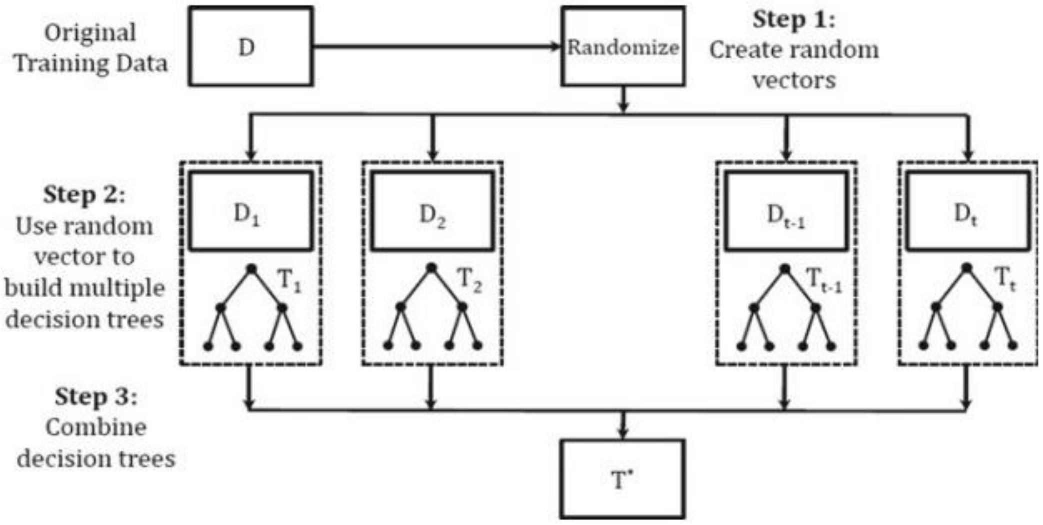
Original Training Data
D
Randomize
Step 1: Create random vectors
Step 2: Use random vector to build multiple decision trees
D$_1$
T$_1$
D$_2$
T$_2$
D$_{t-1}$
T$_{t-1}$
D$_t$
T$_t$
Step 3: Combine decision trees
T*

Fig .2 Representation of random forest

5-Neural networks: The neural network has good self-learning, self-adapting and generalization ability, but it may easily get stuck in a local minimum and has a poor convergence rate.[18] As shown in the figure.3, modeling of input variables as a layer of vertices performed in the network. Then distribution of weight is applied to every connection within the graph. Moreover, the other vertices are placed into separate levels, reflecting the distance from the input nodes [19].

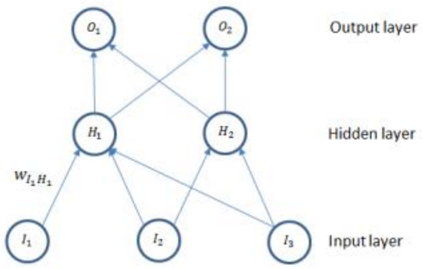

Fig .3 a simple neural network [20].

6-Naïve Bayes: is a supervised learning method that doesn't rely on any attribute. The baseline is the Bayes theorem. Depending on the distribution type. naïve Bayes is a supervised learning method that doesn't rely on any attribute. The baseline is the Bayes theorem. Based on the kind of distribution, the following algorithms are available: Three distributions: Bernoulli, Multinomial, and Gaussian. The Bernoulli distribution is employed in this study to identify fraudulent transactions.[21]

$$P(c|x) = \frac{\mathrm{P(x \mid c)P(c)}}{\mathrm{P(x)}} \qquad (3)$$

$$P(c|x) = P\ (x\ 1 \mid c) \times P\ (\mathrm{x}\ 2 \mid c) \times \cdots \times P\ (\mathrm{x\ n} \mid c) \times P\ (c) \qquad (4)$$

3.3 Performance Metrics

Accuracy, a performance indicator, counts the percentage of examples in a dataset that are properly classified out of all the instances. F1 score aggregates recall and precision into a single number when there is an imbalance between the classes in a binary classification problem; it is especially helpful. Recall is a performance statistic used in classification tasks, sometimes referred to as sensitivity or true positive rate. Precision is a performance indicator in machine learning and statistics that assesses how well a model makes good predictions. It is the proportion of correctly predicted true positives to the total of correctly predicted false positives.

$$\text{Accuracy} = (TN + TP)/(TN + TP + FN + FP) \qquad (5)$$

$$\text{Precision} = TP/TP + FP \qquad (6)$$

$$\text{Recall} = TP/TP + FN \qquad (7)$$

$$\text{Specificity} = (TN/TN + FP) \qquad (8)$$

$$MCC = \frac{\mathrm{TP \times TN - FP \times FN}}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (9)$$

4. Experimental Results

The results collected from Gradient Boosting, AdaBoost, Naïve Bayes, Neural network, SVM, CN7 Rule induction, Logistic Regression, Random Forest, Stochastic gradient Descent, k-nearest Neighbor, Tree, Constant are shown below.

TABLE II
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	AUC	CA	F1	Perc	Recall	MCC
AdaBoost	0.749	0.998	0.998	0.998	0.998	0.451
CN7 Rule induction	0.907	0.998	0.998	0.998	0.998	0.486
Constant	0.455	0.998	0.997	0.996	0.998	0.000
Gradient boosting	0.967	0.997	0.997	0.998	0.997	0.368
KNN	0.719	0.998	0.997	0.996	0.998	0.000
Logistic regression	0.929	0.999	0.998	0.999	0.999	0.583
Naive Bayes	0.962	0.997	0.996	0.996	0.997	-0.002
Neural network	0.892	0.998	0.997	0.996	0.998	0.000
Random forest	0.941	0.998	0.998	0.998	0.998	0.421
SVM	0.763	0.998	0.997	0.996	0.998	0.000
Tree	0.455	0.998	0.997	0.996	0.998	0.000

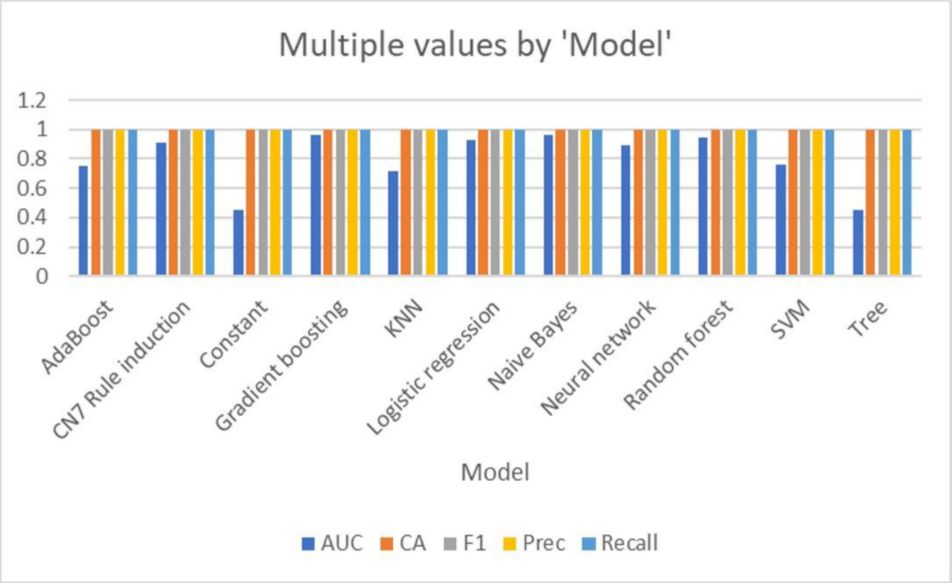

Fig.4 First dataset performance chart with data split

Gradient Boosting leads the pack in machine learning accuracy with an astounding 0.967, closely followed by Naive Bayes, which performs admirably with 0.962. In contrast, the Constant and Tree classifiers show the lowest performance, each obtaining a lower accuracy rate of 0.455.

Table III
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	AUC	CA	F1	Perc	Recall	MCC
AdaBoost	0.725	0.998	0.998	0.998	0.998	0.528
CN7 Rule induction	0.911	0.998	0.998	0.998	0.998	0.498
Constant	0.500	0.998	0.997	0.996	0.998	0.000
Gradient boosting	0.820	0.997	0.997	0.997	0.997	0.281
KNN	0.685	0.998	0.997	0.996	0.998	0.000
Logistic regression	0.903	0.998	0.998	0.997	0.998	0.388
Naive Bayes	0.971	0.996	0.996	0.996	0.996	-0.002
Neural network	0.914	0.998	0.997	0.996	0.998	0.000
Random forest	0.948	0.998	0.997	0.998	0.998	0.316
SVM	0.742	0.998	0.997	0.996	0.998	0.000
Tree	0.500	0.998	0.997	0.996	0.998	0.000

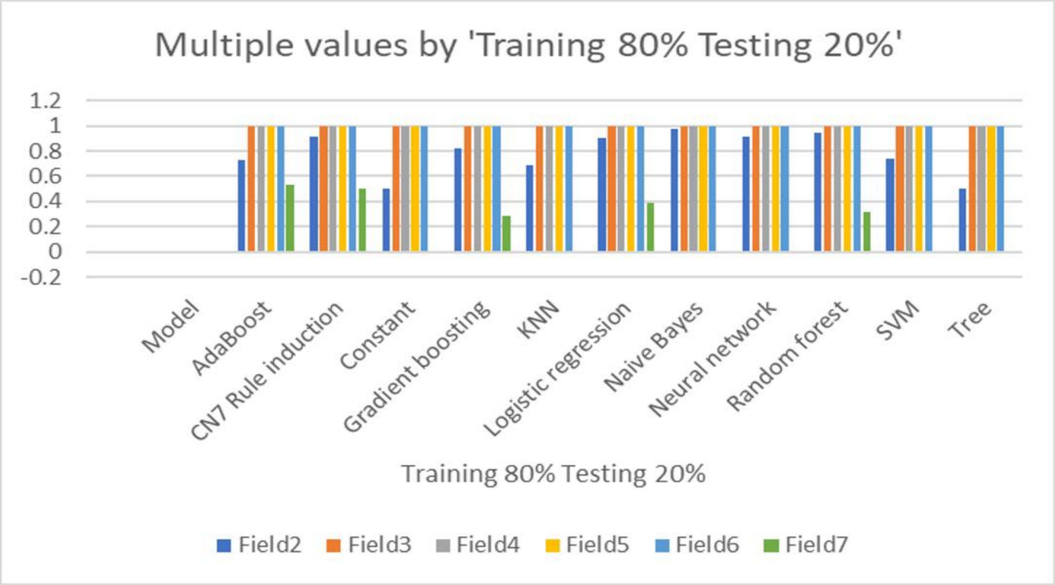

Fig. 5 First dataset performance chart with data split

Naive Bayes is the best-performing model, with an accuracy of 0.971; Random Forest comes in second with 0.948. With a 0.500 accuracy rate, a decision tree is the least accurate model. The following results are from the second data set.

Table IV
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	AUC	CA	F1	Perc	Recall	MCC
Tree	0.414	0.993	0.990	0.986	0.993	0.000
SVM	0.573	0.993	0.990	0.986	0.993	0.000
Random forest	0.957	0.995	0.994	0.995	0.995	0.533
Neural Network	0.853	0.993	0.990	0.986	0.993	0.000
Logistic Regression	0,915	0.995	0.994	0.994	0.995	0.523
Constant	0.414	0.993	0.990	0.986	0.993	0.000
CN2 Rule Induction	0.959	0.992	0.991	0.991	0.992	0.318

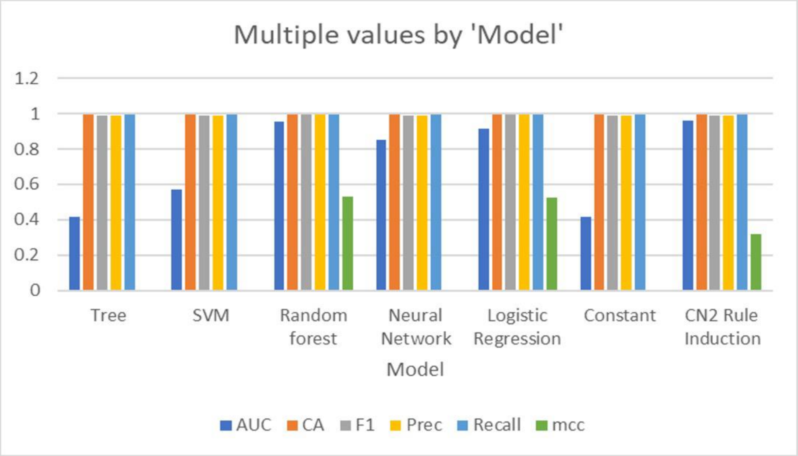

Fig.6. Second dataset performance chart with data split

The highest-performing models are Random Forest (0.957 accuracy) and CN2 Rule Induction (0.959 accuracy), whereas Constant (0.414 accuracy) and Tree (0.414 accuracy) are the lowest-performing models.

Table V
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	AUC	CA	F1	Perc	Recall	MCC
Tree	0.500	0.993	0.989	0.985	0.993	0.000
SVM	0.583	0.993	0.989	0.985	0.993	0.000
Random forest	0.943	0.993	0.991	0.990	0.993	0.286
Neural Network	0.851	0.993	0.989	0.985	0.993	0.000
Logistic Regression	0.903	0.995	0.994	0.994	0.995	0.557
Constant	0.500	0.993	0.989	0.985	0.993	0.000
CN2 Rule Induction	0.980	0.993	0.993	0.992	0.993	0.471

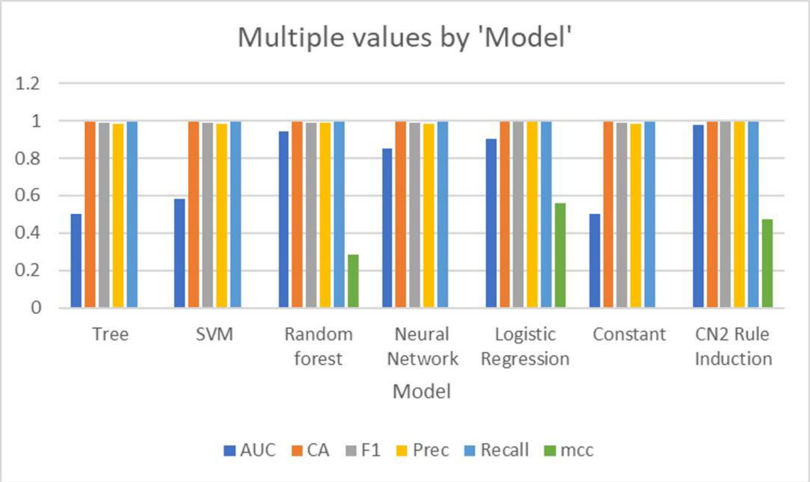

Fig.7. Second dataset performance chart with data split

With an astounding accuracy of 0.980, CN2 Rule Induction is the best-performing model; Random Forest comes in second at 0.943. Constant and Tree models, on the other hand, both score 0.500, which is the lowest Accuracy.

Table VI
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	AUC	CA	F1	Perc	Recall	MCC
AdaBoost	0.854	0.994	0.994	0.995	0.994	0.665
CN7 Rule induction	0.991	0.995	0.995	0.995	0.995	0.663
Constant	0.482	0.992	0.988	0.984	0.992	0.000
Gradient boosting	0.995	0.996	0.996	0.996	0.996	0.767
KNN	0.874	0.993	0.992	0.991	0.993	0.440
Logistic regression	0.944	0.977	0.983	0.993	0.977	0.457
Naive Bayes	0.965	0.985	0.985	0.985	0.985	0.069
Neural network	0.955	0.993	0.990	0.993	0.993	0.388
Random forest	0.991	0.997	0.997	0.997	0.997	0.789
Stochastic gradient Descent	0.515	0.992	0.989	0.992	0.992	0.173
SVM	0.709	0.992	0.988	0.984	0.992	0.000
Tree	0.482	0.992	0.988	0.984	0.992	0.000

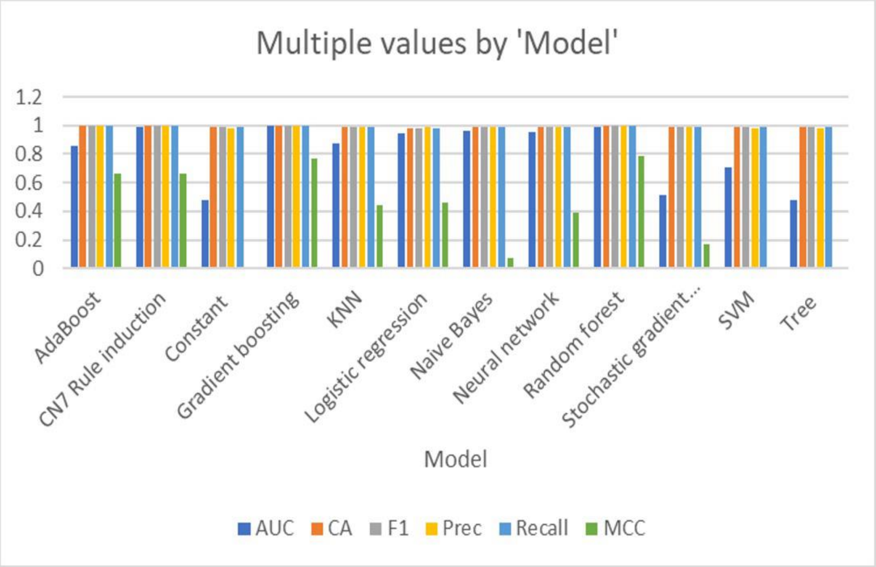

Fig.8. Third dataset performance chart with data split

The first three models demonstrate remarkable Accuracy: Gradient Boosting comes in first with 0.995, closely followed by CN7 Rule Induction and Random Forest, which have excellent Accuracy of 0.991. Conversely, Constant and Tree, with respective scores of 0.482, share the lowest Accuracy.

Table VII
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	AUC	CA	F1	Perc	Recall	MCC
AdaBoost	0.856	0.995	0.995	0.995	0.995	0.682
CN7 Rule induction	0.988	0.995	0.995	0.995	0.995	0.660
Constant	0.500	0.992	0.988	0.984	0.992	0.000
Gradient boosting	0.992	0.996	0.996	0.996	0.996	0.759
KNN	0.845	0.993	0.992	0.992	0.993	0.451
Logistic regression	0.947	0.978	0.984	0.993	0.978	0.482
Naive Bayes	0.962	0.985	0.985	0.985	0.985	0.084
Neural network	0.969	0.993	0.990	0.993	0.993	0.381
Random forest	0.975	0.997	0.997	0.997	0.997	0.784
Stochastic gradient Descent	0.519	0.992	0.989	0.992	0.992	0.195
SVM	0.720	0.992	0.988	0.984	0.992	0.000
Tree	0.500	0.992	0.988	0.984	0.992	0.000

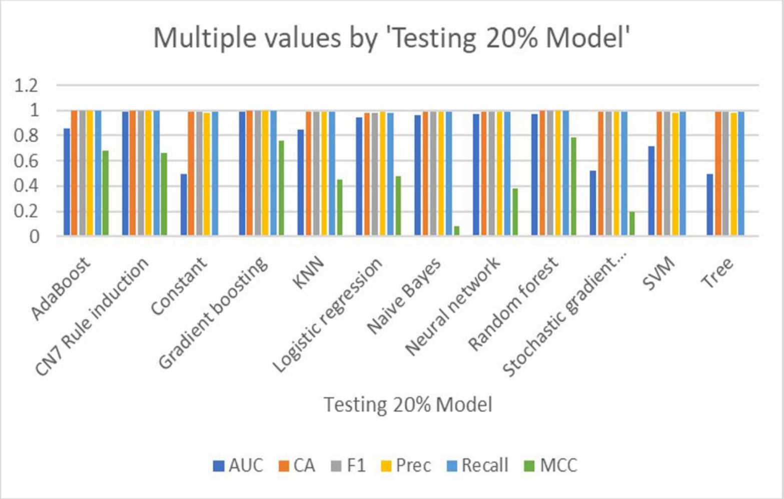

Fig. 9. Third dataset performance chart with data split

Using gradient boosting, the best-performing model achieves an impressive accuracy of 0.992, with random forest coming in second with an accuracy of 0. 975. The model's score is 0.500. however, are the least accurate; these are the constant and three models.

Table VIII
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	AUC	CA	F1	Perc	Recall	MCC
Constant	0.414	0.933	0.989	0.985	0.993	0.000
CN2 Rule Induction	0.959	0.992	0.993	0.992	0.993	0.471
KNN	0.780	0.992	0.989	0.985	0.992	-0.001
Tree	0.414	0.993	0.989	0.985	0.993	0.000
Random Forest	0.957	0.995	0.992	0.994	0.994	0.446
Gradient Boosting	0.993	0.994	0.991	0.990	0.992	0.316
SVM	0.582	0.993	0.989	0.985	0.993	0.000
Logistic Regression	0.915	0.995	0.994	0.994	0.995	0.557
Neural Network	0.853	0.993	0.985	0.985	0.993	0.000

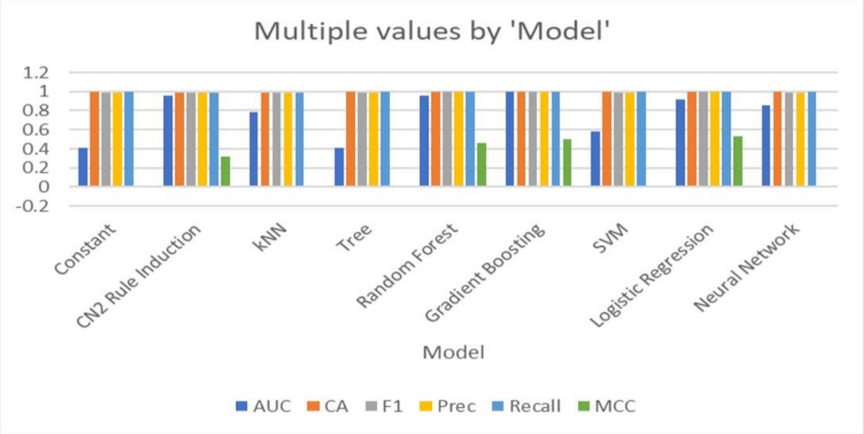

Fig.10. Fourth dataset performance chart with data split

With 0.993 accuracy, Gradient Boosting leads the pack, closely followed by CN2 Rule Induction at 0.959. These top-performing models exhibit outstanding Accuracy. Conversely, at 0.414, Constant and Tree are the models with the lowest Accuracy.

Table IX
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	AUC	CA	F1	Prec	Recall	MCC
Constant	0.500	0.993	0.989	0.985	0.993	0.000
CN2 Rule Induction	0.980	0.993	0.993	0.992	0.993	0.471
kNN	0.679	0.992	0.989	0.985	0.992	-0.001
Tree	0.500	0.993	0.989	0.985	0.993	0.000
Random Forest	0.976	0.994	0.992	0.994	0.994	0.446
Gradient Boosting	0.977	0.992	0.991	0.990	0.992	0.316
SVM	0.612	0.993	0.989	0.985	0.993	0.000
Logistic Regression	0.903	0.995	0.994	0.994	0.995	0.557
Neural Network	0.851	0.993	0.989	0.985	0.993	0.000

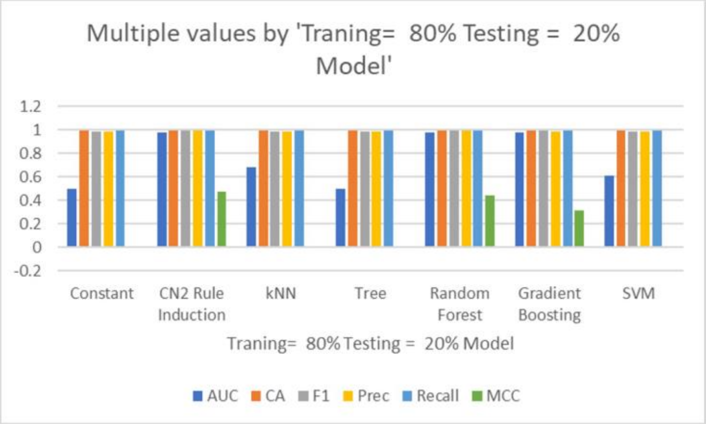

Fig.11. Fourth dataset performance chart with data split

The most accurate model, CN2 Rule Induction, reaches a remarkable accuracy of 0.980. Gradient Boosting comes in second, with an accuracy of 0.977, very close behind.

Table X
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	AUC	CA	F1	Perc	Recall	MCC
Constant	0.488	0.993	0.990	0.987	0.993	0.000
CN7 Rule induction	0.985	0.996	0.996	0.996	0.996	0.664
KNN	0.812	0.994	0.992	0.992	0.994	0.358
Tree	0.488	0.993	0.990	0.987	0.993	0.000
Random forest	0.984	0.997	0.997	0.997	0.997	0.776
Gradient boosting	0.997	0.997	0.997	0.997	0.997	0.772
SVM	0.746	0.993	0.990	0.987	0.993	0.000
Logistic regression	0.926	0.979	0.985	0.994	0.979	0.417
Naive Bayes	0.975	0.987	0.988	0.988	0.987	0.105
AdaBoost	0.830	0.996	0.996	0.996	0.996	0.680
Neural network	0.966	0.994	0.992	0.994	0.994	0.364
Stochastic gradient Descent	0.515	0.994	0.990	0.994	0.994	0.171

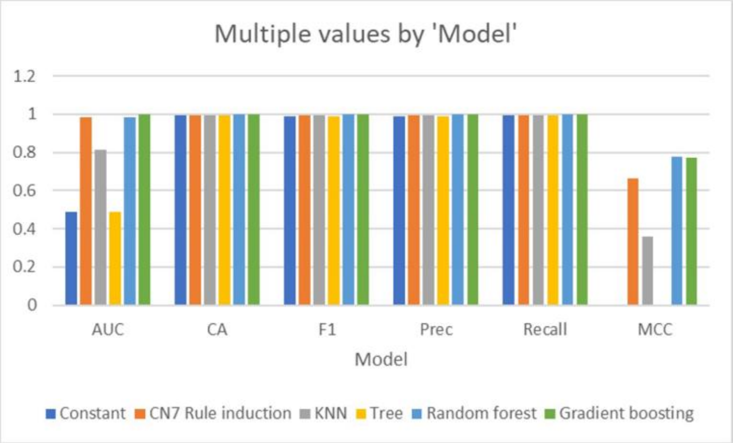

Fig. 12. Fifth dataset performance chart with data split

Gradient Boosting (0.997) and CN7 Rule Induction (0.985) are the two best methods in terms of Accuracy. Random Forest follows closely, obtaining an accuracy of 0.984. However, with respective accuracy values of 0.488, the Constant and Tree models show the lowest Accuracy.

Table XI
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	AUC	CA	F1	Perc	Recall	MCC
Constant	0.500	0.993	0.990	0.986	0.993	0.000
CN7 Rule induction	0.977	0.995	0.995	0.995	0.995	0.650
KNN	0.853	0.994	0.992	0.992	0.994	0.360
Tree	0.500	0.993	0.990	0.986	0.993	0.000
Random forest	0.959	0.997	0.997	0.997	0.997	0.784
Gradient boosting	0.997	0.997	0.997	0.997	0.997	0.797
SVM	0.768	0.993	0.990	0.986	0.993	0.000
Logistic regression	0.890	0.982	0.986	0.993	0.982	0.390
Naive Bayes	0.976	0.988	0.988	0.988	0.988	0.088
AdaBoost	0.835	0.996	0.996	0.996	0.996	0.677
Neural network	0.939	0.994	0.992	0.994	0.994	0.348
Stochastic gradient Descent	0.500	0.993	0.990	0.986	0.993	0.000

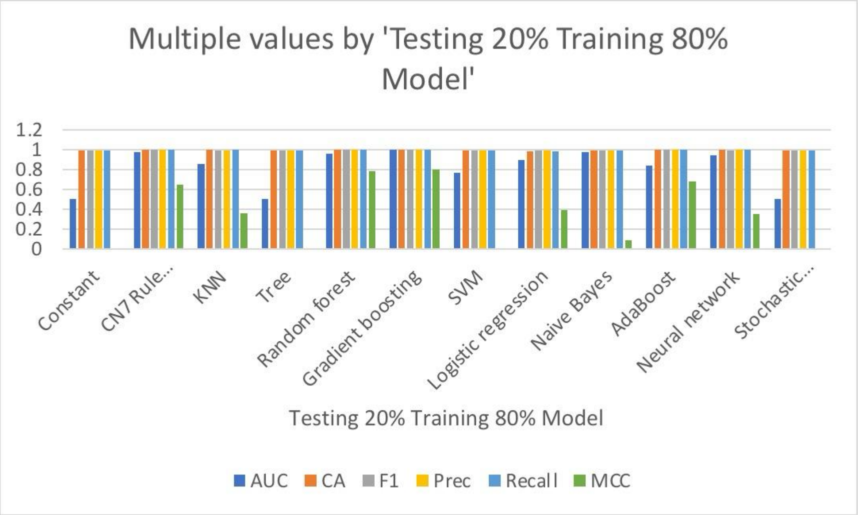

Fig. 13. Fifth dataset performance chart with data split

Gradient Boosting and CN7 Rule Induction, the two best performers, both have the highest Accuracy 0.977; Naive Bayes, coming a close second, achieved the second-highest Accuracy of 0.976. Conversely, the Constant, Tree, and Stochastic Gradient Descent models all had the lowest Accuracy, with a score of 0.500.

Table XII
STATISTICS OF ALGORITHMS WITH 10 K-FOLD

Model	AUC	CA	F1	perc	recall	MCC
SVM	0.605	0.993	0.989	0.985	0.993	0.000
Tree	0.500	0.993	0.989	0.985	0.993	0.000
Constant	0.500	0.993	0.989	0.985	0.993	0.000
Naive Bayes	0.936	0.982	0.984	0.987	0.982	0.069
KNN	0.761	0.992	0.989	0.985	0.992	- 0.002
Neural Network	0.894	0.993	0.989	0.985	0.993	0.000
CN2 Rule induction	0.980	0.993	0.993	0.993	0.993	0.479
Random Forest	0.958	0.995	0.993	0.994	0.995	0.476
Gradient boosting	0.984	0.993	0.993	0.992	0.993	0.441

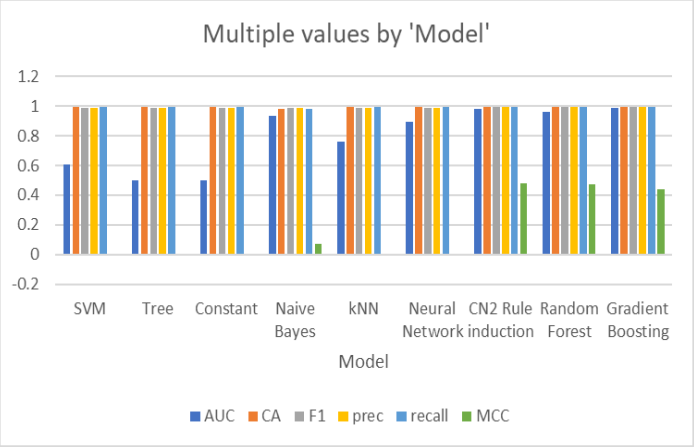

Fig.14. Sixth dataset performance chart with data split

Using k-fold, Gradient Boosting, and CN2 Rule Induction are the best-performing models, with 0.984 and 0.980 accuracy, respectively. Conversely, Constant is the least accurate model, with an accuracy of 0.500.

Table XIII
STATISTICS OF ALGORITHMS WITH 80/20 DATA SPLIT

Model	AUC	CA	F1	perc	recall	MCC
SVM	0.575	0.992	0.989	0.985	0.993	0.000
Tree	0.500	0.993	0.989	0.985	0.993	0.000
Constant	0.500	0.993	0.989	0.985	0.993	0.000
Naive Bayes	0.936	0.982	0.984	0.987	0.982	0.069
KNN	0.761	0.992	0.989	0.985	0.992	-0.002
Neural Network	0.894	0.993	0.989	0.985	0.993	0.000
CN2 Rule induction	0.955	0.992	0.992	0.991	0.992	0.400
Random Forest	0.968	0.994	0.993	0.993	0.994	0.477
Gradient Boosting	0.969	0.993	0.992	0.992	0.993	0.420

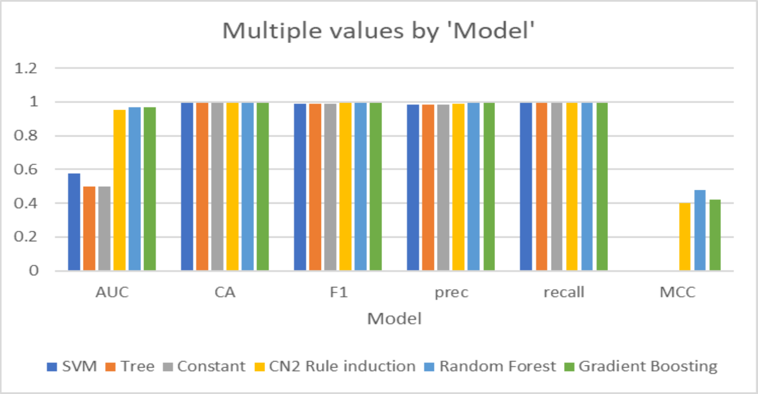

Fig.15. Sixth dataset performance chart with data split

With gradient boosting, the best-performing model achieves an Accuracy of 0.969. Random forest comes in second, with the second-highest Accuracy of 0.968. Conversely, with scores of 0.500, the constant and Tree models show the least Accuracy.

5. Conclusion

Machine learning is a powerful tool in detecting and preventing online payment fraud. It can analyze large amounts of data, identify patterns, and make accurate predictions. By leveraging features like transaction amounts, locations, timestamps, user behavior, and device information, machine-learning models can identify suspicious patterns and flag fraudulent transactions in real-time. However, they can produce false positives or negatives, so combining machine learning and human expertise is crucial. The success of machine learning models depends on the quality and relevance of training data.

References

[1] 8ir5Sakharova, I. (2012, June). Payment card fraud: Challenges and solutions. In 2012 IEEE international conference on intelligence and security informatics (pp. 227-234). IEEE

[2] Almazroi, A. A., & Ayub, N. (2023). Online Payment Fraud Detection Model Using Machine Learning Techniques. IEEE Access, 11, 137188-137203

[3] El Naqa, I., & Murphy, M. J. (2015). What is machine learning? (pp. 3-11). Springer International Publishing.

[4] Minastireanu, E. A., & Mesnita, G. (2019). An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection. Informatica Economica, 23(1).

[5] Nasr, M. H., Farrag, M. H., & Nasr, M. M. (2022). A Proposed Fraud Detection Model based on e-Payments Attributes a Case Study in Egyptian e-Payment Gateway. International Journal of Advanced Computer Science and Applications, 13(5).

[6] Fang, Y., Zhang, Y., & Huang, C. (2019). Credit Card Fraud Detection Based on Machine Learning. Computers, Materials & Continua, 61(1).

[7] Mijwil, M. M., & Salem, I. E. (2020). Credit card fraud detection in payment using machine learning classifiers. Asian Journal of Computer and Information Systems (ISSN: 2321–5658), 8(4).

[8] Adepoju, O., Wosowei, J., & Jaiman, H. (2019, October). Comparative evaluation of credit card fraud detection using machine learning techniques. In 2019 Global Conference for Advancement in Technology (GCAT) (pp. 1-6). IEEE.

[9] Isabella, S. J., Srinivasan, S., & Suseendran, G. (2020). An efficient study of fraud detection system using Ml techniques. Intelligent Computing and Innovation on Data Science, 59.

[10] Pumsirirat, A., & Liu, Y. (2018). Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. International Journal of advanced computer science and applications, 9(1).

[11] Corballis, M. C., & Nagourney, B. A. (1978). Latency to categorize disoriented alphanumeric characters as letters or digits. Canadian Journal of Psychology/Revue canadienne de psychologie, 32(3), 186.

[12] Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in neurorobotics, 7, 21.

[13] Asim, M., & Zakria, M. (2020). Advanced kNN: A Mature Machine Learning Series. arXiv preprint arXiv:2003.00415.

[14] Ngai, E. W., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision support systems, 50(3), 559-569.

[15] Nusinovici, S., Tham, Y. C., Yan, M. Y. C., Ting, D. S. W., Li, J., Sabanayagam, C., ... & Cheng, C. Y. (2020). Logistic regression was as good as machine learning for predicting major chronic diseases. Journal of clinical epidemiology, 122, 56-69.

[16] Ravisankar, P., Ravi, V., Rao, G. R., & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. Decision support systems, 50(2), 491-500.

[17] Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. Expert systems with applications, 134, 93-101.

[18] Ding, S., Su, C., & Yu, J. (2011). An optimizing BP neural network algorithm based on genetic algorithm. Artificial intelligence review, 36, 153-162.

[19] Kirkos, E., Spathis, C., & Manolopoulos, Y. (2007). Data mining techniques for the detection of fraudulent financial statements. Expert systems with applications, 32(4), 995-1003.

[20] West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: a comprehensive review. Computers & security, 57, 47-66.

[21] Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. Knowledge-Based Systems, 192, 105361.