Green Intelligent Systems and Applications

Research Article
Volume 5(1), 2025, 40–50
https://doi.org/10.53623/gisa.v5i1.552

Fraud Classification in Online Payments Using Supervised Machine Learning Algorithms

Arda Surya Editya*, Moch. Machlul Alamin, Anggay Lury Pramana, Neny Kurniati

Departement of Informatics, Universitas Nahdlatul Ulama Sidoarjo, Indonesia

*Correspondence: ardasurya.tif@unusida.ac.id

SUBMITTED: 29 November 2025; REVISED: 4 March 2025; ACCEPTED: 15 March 2025

ABSTRACT: Online payment systems have become a cornerstone of modern financial transactions, providing convenience and efficiency. However, the rise of such systems has also led to an increase in fraudulent activities, posing significant risks to users and service providers. This research focused on optimizing the classification of fraudulent transactions in online payment systems using supervised machine learning algorithms. This study explored the performance of several widely used algorithms, including Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting Tree, and Support Vector Machine (SVM). A comprehensive dataset of online payment transactions was used to evaluate the effectiveness of these algorithms in identifying fraudulent activities. Various performance metrics, such as accuracy, precision, and F1 score, were employed to assess and compare classification capabilities. In addition, feature engineering and data preprocessing techniques were applied to improve the models' predictive performance. The results demonstrated that, while each algorithm had its strengths, ensemble-based methods like Gradient Boosting Tree outperformed others in terms of classification accuracy and robustness. The findings highlighted the importance of selecting appropriate machine learning algorithms and fine-tuning their parameters to achieve optimal fraud detection in online payment systems. This study provides valuable insights for financial institutions and developers to enhance security measures and mitigate fraud risks in digital payment platforms.

KEYWORDS: Fraud; classification; machine learning.

1. Introduction

The growth of digital payment systems has transformed how financial transactions are conducted, providing users with unmatched convenience and accessibility. However, this rapid adoption has also been accompanied by a surge in fraudulent activities, which pose a significant challenge for financial institutions and payment service providers [1]. Fraudulent transactions not only result in financial losses but also erode user trust and can lead to regulatory penalties. Consequently, detecting and preventing fraudulent online payments has become a critical priority. Traditional rule-based systems for fraud detection are effective to some extent; however, they often struggle to adapt to the evolving tactics employed by fraudsters [2]. These systems are typically rigid, require constant updates, and can generate a high rate of false positives, leading to unnecessary delays or the rejection of legitimate transactions. Machine

learning (ML), which can learn from large datasets and detect patterns indicative of fraud, has emerged as a powerful tool to address this issue [3].

This research focused on optimizing fraud classification in online payment systems using supervised machine learning algorithms. Supervised learning provided a framework for training models to distinguish between fraudulent and legitimate transactions based on labeled data [4]. This study evaluated the performance of various popular algorithms, including Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting Tree, and Support Vector Machine (SVM), to identify the most effective method for fraud detection. Through a comparative analysis of these algorithms, this research aimed to determine their strengths, weaknesses, and suitability for different aspects of fraud detection, such as precision, recall, and the ability to handle imbalanced datasets. The results of this study offered valuable insights into developing strong, scalable, and efficient fraud detection systems that can adapt to the ever-changing landscape of online payment fraud.

2. Previous Work

The application of machine learning techniques in fraud detection has been extensively studied, reflecting its growing importance in safeguarding online payment systems. Many researchers have investigated various algorithms and methods to detect fraudulent activities with improved accuracy and efficiency. Early studies primarily focused on traditional statistical methods; however, recent advancements in supervised learning techniques have significantly enhanced the ability to classify fraudulent transactions. This section reviews the relevant literature and highlights key methodologies, algorithmic developments, and challenges addressed by previous studies.

2.1. Fraud detection in banking system.

Research into fraud detection in banking systems was conducted by Mehdipour et al. They found that advancements in banking IT have led to increased fraud, necessitating further research on technological aspects and responses from authorities to improve detection and mitigation [5]. Furthermore, Ghosh et al. demonstrated that ISOMAP improves fraud detection by reducing data complexity and preserving data structure, leading to higher accuracy and fewer false positives in various sectors. ISOMAP (Isometric Mapping) is a non-linear dimensionality reduction technique used to compute a low-dimensional embedding from a set of high-dimensional data points [6].

The literature on fraud detection includes several studies that have implemented artificial intelligence to detect fraud in banking systems. Dash et al. applied AI and machine learning methods to detect fraud in the banking and financial sector. Their study utilized neural networks, which proved to be more effective than traditional approaches. This research emphasized the significance of data management in developing advanced fraud detection systems [7].

In contrast, Ayeni et al. examined fraud detection in eight international banks in Nigeria using surveys and data analysis methods such as SPSS and SEM-PLS. Their results demonstrated that AI improves fraud awareness and transaction security. This study recommended that banks adopt AI technologies and collaborate with cybersecurity experts to continuously enhance fraud prevention measures [8].

2.2. Machine learning in fraud detection.

The use of machine learning in fraud detection, especially in online transactions and banking systems, has been studied by several researchers, such as Balaji et al.. In their study, they stated that the banking industry requires robust detection systems to combat fraud and maintain trust. Traditional systems often fail against complex fraud schemes. Their study presented an all-encompassing approach that incorporated machine learning, big data analytics, and essential management components. It emphasized real-time monitoring and automatic alerts, ensuring legal compliance and effective fraud prevention in financial institutions [9].

Furthermore, Shah and Mehta evaluated six machine learning techniques for credit card fraud detection using confusion matrices. Metrics such as accuracy, precision, recall, specificity, misclassification rate, and F1 score demonstrated high efficacy. They recommended using multiple techniques for improved fraud detection [10]. Several machine learning methods have been commonly used to detect fraud, such as Naïve Bayes. The application of Naïve Bayes in fraud detection was studied by Aladakatti et al.. Their results demonstrated that machine learning algorithms can effectively detect fraudulent activities by analyzing sufficient transaction data.

In this study, we utilized several machine learning techniques, including Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes, Decision Tree, and Random Forest, to identify patterns and anomalies indicative of fraud. The Random Forest classifier demonstrated superior performance, achieving an accuracy rate of 99.94%. This finding underscored the potential of Random Forest as a powerful tool in fraud detection, offering reliable and accurate results compared to other classifiers [10]. In addition, Lochan et al. developed a hybrid credit card fraud detection technique. The results of their study demonstrated that the hybrid technique exhibited better accuracy, precision, and recall than k-means clustering [11].

3. Materials and Methods

This section describes the proposed method of this research for detecting fraud in online payments. As shown in Figure 1, the first step was gathering the dataset from Abbasi and Shah (2022) [12]. After that, we preprocessed the data, including feature selection and data balancing. Once the dataset was preprocessed, the next step was training. In this method, we used several machine learning techniques, such as Naïve Bayes, Logistic Regression, Support Vector Machine (SVM), Random Forest, and Gradient Boosted Trees. After the training process, the next step was evaluation to determine the best method for classifying fraudulent online payments. In the evaluation step, we used several parameters, such as accuracy, precision, recall, and F1-score.

3.1. Data preprocessing.

This study uses a dataset from Abbasi and Shah (2022), which is available at (https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection). his dataset contains 6,362,261 records [13]. The data composition is shown in Table 1.

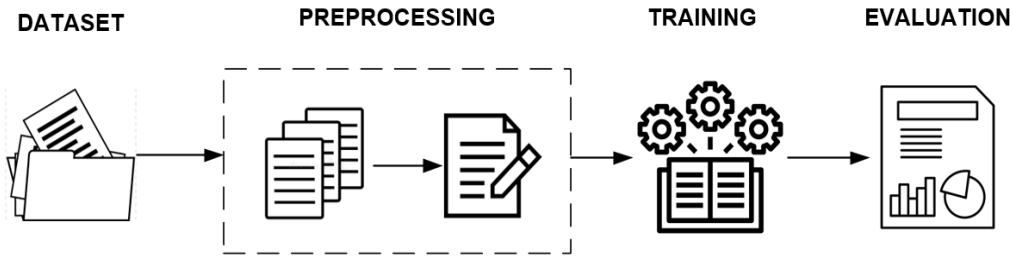

Figure 1. The research diagram.

Based on Table 1, we observe an imbalance in the dataset, which can affect the performance of machine learning models during training. Therefore, we need to reduce the total data and create a balanced dataset. To achieve this, we used an undersampling technique.

Table 1. Composition of dataset.

Class	Number of Data
Normal	6,2361,119
Fraud	1,142

The undersampling technique is a method used to reduce the majority class by removing samples to match the size of the minority class. Table 2 shows the dataset after applying the undersampling technique. After obtaining a balanced dataset, the next step is to show the dataset parameters, as presented in Table 3.

Table 2. Composition of dataset after undersampling.

Class	Number of Data
Normal	1,514
Fraud	1,142

Table 3. Dataset parameter.

Parameter	Description
step	represents a time unit where each step is equivalent to one hour.
type transaction	categories of online transactions
amount of money	the transaction value
name of sender	the customer initiating the transaction
oldbalanceOrig	the balance prior to the transaction
newbalanceOrig	the balance following the transaction
nameDest	the party receiving the transaction
oldbalanceDest	the recipient's balance prior to the transaction
newbalanceDest	the recipient's updated balance after the transaction
isFraud	fraudulent transaction

Table 3 shows the parameters of the dataset. However, in the data preprocessing step, we did not use all the parameters. We selected only the parameters relevant for classification. The final parameters included transaction type, transaction amount, oldbalanceOrig, newbalanceOrig, oldbalanceDest, and newbalanceDest. The isFraud parameter was used as the target variable (y value).

3.2. Naïve bayes.

Naïve Bayes is a supervised learning algorithm based on Bayes' Theorem, which provides a probabilistic approach for classifying data [14]. It is particularly effective for tasks such as text classification, spam filtering, sentiment analysis, and fraud detection. Despite its simplicity,

Naïve Bayes often delivers strong performance, especially on large datasets. Naïve Bayes assumes that all features are independently distributed given the class label. Although this assumption is rarely accurate in real-world scenarios, it greatly simplifies computations and improves the algorithm's computational efficiency. Equation 1 presents the formula for Naïve Bayes.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (1)$$

Bayes Theorem is the mathematical principle underpinning the Naive Bayes classifier, providing a probabilistic framework for decision-making. It describes the relationship between the conditional probabilities of events and allows the computation of the posterior probability P(C|X), which is the probability of a class C given the observed data X. where P(C) represents the prior probability of the class, P(X|C) is the likelihood of the data given the class, and P(X) is the evidence or the overall probability of the data [14]. In Naive Bayes, the posterior probability is calculated for each class, and the class with the highest posterior probability is chosen as the prediction. The method's assumption of feature independence simplifies P(X|C) into a product of probabilities for individual features, significantly reducing computational complexity and making the model efficient.

3.3. Decision tree.

Decision Tree is a supervised machine learning model that classifies data by recursively splitting it into subsets based on feature values, forming a tree-like structure. It uses recursive partitioning to classify data or predict continuous values. The splitting criteria at each node are typically determined using metrics such as Gini Impurity or Entropy, as shown in Equation 2.

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

Where p_i is the proportion of class *i* in the node. A Gini value of 0 indicates perfect purity. [15]. Furthermore there is a entropy to used in information gain, this can shown in equation 3.

$$H = -\sum_{i=1}^{n} p_i \log_2(pi) \quad (3)$$

Entropy measures the randomness in the dataset. Lower entropy indicates higher purity. Furthermore, the Decision Tree splits at each node by maximizing Information Gain (IG).

$$IG = H_{parent} - \sum_{j=1}^{k} \frac{N_j}{N} H_{childj} \quad (4)$$

where H_{parent} is the entropy of the parent node, H_{childj} is the entropy of the j-th child node, N_j is the number of samples in the j-th child, and N is the total samples in the parent node. The algorithm terminates when all nodes are pure or meet a stopping criterion (e.g., maximum depth, minimum samples per node). Decision Trees are interpretable but prone to overfitting,

which can be mitigated using pruning techniques or ensemble methods like Random Forest [15].

3.4. Support vector machine (SVM).

Support Vector Machine (SVM) is a supervised machine learning algorithm commonly employed for both classification and regression tasks [16]. Its primary strength lies in its ability to find a hyperplane that best separates data points into distinct classes, making it particularly effective for high-dimensional and complex datasets. Developed based on statistical learning theory, SVM has become a cornerstone in machine learning for tasks such as image recognition, text classification, and fraud detection. The goal of SVM is to find the optimal hyperplane that maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors. A larger margin reduces the generalization error and enhances model performance. For a binary classification problem, given a training dataset with mmm samples, where X={x1,x2,...,xm}X = \{x_1, x_2, ..., x_m\}X={x1,x2,...,xm} represents the feature vectors and Y={y1,y2,...,ym}Y = \{y_1, y_2, ..., y_m\}Y={y1,y2,...,ym}, yi∈{−1,1}y_i \in \{-1, 1\}yi∈{−1,1} denotes the class labels [16], Equation 3 shows how SVM solves optimization problem using objective function.

$$min_{w,b}\frac{1}{2}\|w\|^2 \tag{5}$$

In the context of Support Vector Machine (SVM), the constraints ensure that the model finds a hyperplane that correctly classifies the training data (or does so with minimal violations for non-linearly separable data). In optimization problems, there is subject to the constraints that specifies the conditions or rules that must be satisfied while finding the solution to the optimization objective. Constraints define the permissible set of solutions, narrowing down the search space to ensure that the optimization respects the problem's requirements. Equation 4 shows the subject to the constraints.

$$y_i(w.x_i + b) \geq 1 \quad \forall_i = 1, \ldots, m \tag{6}$$

Where y_i is the label of the i-th data point, $w.x_i+b$ is the signed distance of the i-th data point from the hyperplane and the l is the margin threshold for correctly classified points.

3.5. Random forest.

Random Forest is a supervised learning algorithm that utilizes ensemble techniques to enhance the performance of decision trees. It is widely used for both classification and regression tasks due to its robustness, accuracy, and ability to handle large datasets with high-dimensional features [17]. Random Forest constructs multiple decision trees during training and combines their outputs to produce more stable and accurate predictions. Its foundation lies in the principles of bagging (Bootstrap Aggregating) and random feature selection, which help reduce overfitting and improve generalization. Random Forest is a versatile and powerful algorithm that capitalizes on the strengths of decision trees while mitigating their weaknesses through ensemble learning. Its ability to handle non-linear relationships, resistance to noise, and scalability make it an essential tool in machine learning. With proper hyperparameter tuning,

Random Forest delivers competitive performance across a wide range of real-world applications, particularly for high-dimensional and complex datasets.

3.6. Gradient boosted trees.

Gradient Boosted Trees (GBT) is a supervised machine learning algorithm that combines the strengths of decision trees and gradient boosting to achieve high predictive accuracy [18]. By sequentially constructing an ensemble of weak learners (typically decision trees), GBT minimizes the errors of previous models using gradient descent in a functional space. This iterative approach enables GBT to capture complex patterns in data, making it a popular choice for both classification and regression tasks. Gradient Boosted Trees integrate the simplicity of decision trees with the power of gradient-based optimization, making them one of the most effective algorithms for structured data [18]. By sequentially reducing errors through boosting, GBT delivers high predictive performance across various domains. However, due to its computational demands and sensitivity to hyperparameters, careful implementation and evaluation are crucial for achieving optimal results.

4. Results and Discussion

This section presents the results and analysis of fraud classification optimization in online payment systems using supervised machine learning algorithms. The study evaluates multiple models based on accuracy, precision, recall, and F1-score to assess their effectiveness in detecting fraudulent transactions. A comparative analysis highlights performance differences among the models, while feature importance insights enhance interpretability and model reliability.

4.1. Environment experiment.

The experiments in this study were conducted on a computer equipped with high-performance hardware to ensure efficient data processing and model training. The system specifications include an AMD Ryzen 5 processor with a clock speed of 3.0 GHz, providing sufficient computational power for handling complex machine learning tasks. The machine is further supported by 32 GB of RAM, allowing for smooth data handling even with large datasets and minimizing memory bottlenecks during processing. For computational processing, the computer is equipped with an NVIDIA GTX 1650 Ti GPU. Although not a high-end model, this GPU provides adequate support for accelerating tasks such as data visualization and certain parallelized computations during model training. This hardware setup was selected to balance performance and cost efficiency, ensuring reliable experimentation within the study's constraints.

RapidMiner was utilized as the primary software for data preprocessing, model training, and evaluation. This platform offers an intuitive interface and a wide range of tools for machine learning workflows, making it well-suited for the study's objectives. Its compatibility with the chosen hardware ensured seamless integration, enabling efficient execution of all experimental procedures. The data used in this study was divided into two sets: 70% for training and 30% for testing. This partitioning was determined based on the dataset's characteristics and the need to ensure that the model effectively adapts to the data. Proper data splitting helps optimize model performance by balancing training stability and evaluation reliability.

4.2. Result.

This subsection presents the results obtained from the experiments, highlighting the performance of supervised machine learning algorithms in classifying online payment fraud. Key metrics such as accuracy, precision, recall, and F1-score are analyzed to evaluate each model's effectiveness. The findings are further compared to identify the most optimal algorithm for fraud detection based on the dataset used in this study. Table 1 presents the experimental results, including the accuracy, precision, recall, and F1-score for each algorithm.

Table 1. Results of each method.

Method	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	73.7	93.7	41.7	57.5
Decision Tree	57.0	60.2	30.3	40.7
Random Forest	65.9	100	20.5	34.0
Gradient Boost	**77.5**	**100**	**47.5**	**64.4**
Support Vector Machine (SVM)	57.0	60.2	30.4	40.2

Table 1 presents a performance comparison of five machine learning methods for fraud classification: Naïve Bayes, Decision Tree, Random Forest, Gradient Boost, and Support Vector Machine (SVM). Each algorithm's performance is evaluated using accuracy, precision, recall, and F1-score, highlighting their strengths and weaknesses in handling fraud detection tasks. Gradient Boost achieves the highest accuracy (77.5%) among all methods, indicating its effectiveness in correctly classifying the majority of transactions. Its perfect precision (100%) implies that every transaction predicted as fraudulent is indeed fraudulent. However, its recall (47.5%) reveals that it misses a significant portion of actual fraud cases, which is critical in fraud detection. Its balanced F1-score (64.4%) reflects a trade-off between precision and recall.

Furthermore, Naïve Bayes performs moderately well, with an accuracy of 73.7%. Its high precision (93.7%) demonstrates reliability in fraud predictions, but its recall (41.7%) is lower, suggesting that many fraudulent transactions go undetected. The F1-score (57.5%) indicates that Naïve Bayes is reasonably balanced but less effective than Gradient Boost in fraud classification. On the other hand, Decision Tree and SVM perform similarly, with accuracies of 57.0%, and closely aligned precision, recall, and F1-scores. Their lower recall (~30%) compared to precision (~60%) suggests that these methods are conservative in identifying fraud, favoring precision over sensitivity. Consequently, their F1-scores (~40%) indicate weaker overall performance compared to other methods.

Finally, Random Forest, while achieving perfect precision (100%), struggles with a low recall (20.5%), resulting in a lower F1-score (34%). This indicates an overemphasis on precision at the cost of missing a substantial number of fraudulent cases. While Random Forest ensures no false fraud alarms, its limited recall makes it unsuitable for scenarios requiring high fraud detection rates.

The ROC curve illustrates the performance of five machine learning models in distinguishing between fraudulent and non-fraudulent transactions. Gradient Boost (green curve) demonstrates the best performance, with its curve closely approaching the top-left corner, signifying a high true positive rate and low false positive rate. Naïve Bayes (blue curve) and Random Forest (red curve) also perform well but are slightly less optimal, indicating effective but slightly less precise classification capabilities compared to Gradient Boost.

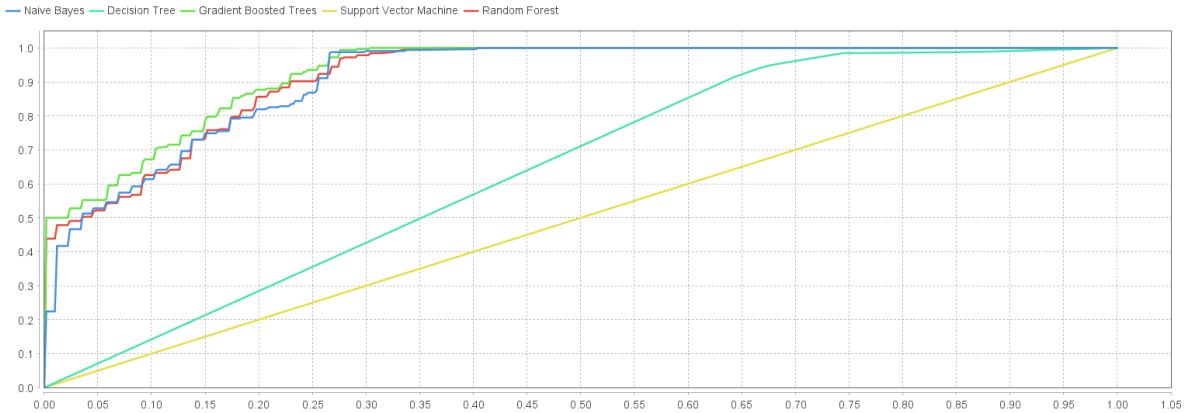

Figure 2. ROC curve comparison.

In contrast, Decision Tree and SVM (yellow curves) show a near-diagonal pattern, representing performance close to random guessing. Their inability to achieve a significant separation between classes suggests limited utility for fraud detection in this dataset. Overall, Gradient Boost emerges as the most reliable model, followed by Naïve Bayes and Random Forest, while Decision Tree and SVM fall behind in classification accuracy.

5. Conclusions

In conclusion, this study evaluates five machine learning models for online payment fraud detection. Gradient Boost demonstrated the highest accuracy and a strong balance between precision and recall, making it the most effective method for this dataset. Naïve Bayes and Random Forest also showed competitive performance but were slightly less optimal. On the other hand, Decision Tree and SVM struggled with classification, exhibiting lower accuracy and weak recall, making them less suitable for fraud detection tasks. Future work could focus on improving recall for models with high precision, such as Random Forest and Gradient Boost, to enhance their ability to detect more fraudulent transactions. Additionally, exploring ensemble techniques, advanced hyperparameter optimization, and incorporating real-time data streams could further boost model performance and applicability in dynamic fraud detection systems.

Author Contribution

This section outlines the contributions of each author to the study, ensuring transparency and accountability in the research process. Specific roles of author as follows: Conceptualization: Arda Surya Editya; Methodology: Arda Surya Editya; Data Collection: Moch. Machlul Alamin; Data Analysis: Arda Surya Editya and Anggay Lury Pramana; Writing: Neny Kurniati and Moch Machlul Alamin; Supervision: Arda Surya Editya.

Competing Interest

All authors should disclose any financial, personal, or professional relationships that might influence or appear to influence their research.

References

[1] Mangala, D.; Soni, L. (2023). A systematic literature review on frauds in the banking sector. *Journal of Financial Crime*, *30*, 285–301. https://doi.org/10.1108/JFC-12-2021-0263.

[2] Bello, H.O.; Ige, A.B.; Ameyaw, M.N. (2024). Adaptive machine learning models: concepts for real-time financial fraud prevention in dynamic environments. *World Journal of Advanced Engineering Technology and Sciences*, *12*, 021–034. http://doi.org/10.30574/wjaets.2024.12.2.0266.

[3] Sharma, R.; Mehta, K.; Sharma, P. (2024). Role of artificial intelligence and machine learning in fraud detection and prevention. Risks and Challenges of AI-Driven Finance: Bias, Ethics, and Security, IGI Global, 90–120.

[4] Dhankhad, S.; Mohammed, E.; Far, B. (2018). Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 122–125.

[5] Mehdipour, F.; Babenkov, E.; Hewage, U.H.W.A.; Aharari, A. (2023). Banking fraud identification and prevention. Proceedings of the 27th International Conference on Circuits, Systems, Communications and Computers (CSCC), Rhodes (Rodos) Island, Greece, 1–6. http://doi.org/10.1109/CSCC58962.2023.00019.

[6] Ghosh, S.; Kumar, J.; Pangotra, T. (2023). Fraud detection system analysis. Proceedings of the 14th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 1–4. http://doi.org/10.1109/ICCCNT56998.2023.10307508.

[7] Dash, S.; Das, S.; Sivasubramanian, S.; Sundaram, N.K.; H.K.G.; Sathish, T. (2023). Developing AI-based fraud detection systems for banking and finance. Proceedings of the 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 891–897. http://doi.org/10.1109/ICIRCA57980.2023.10220838.

[8] Ayeni, T.J.; Durotoye, E.O.; Eriabie, S. (2024). Adoption of artificial intelligence for fraud detection in deposit money banks in Nigeria. Proceedings of the International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG), Omu-Aran, Nigeria, 1–5. http://doi.org/10.1109/SEB4SDG60871.2024.10630329.

[9] Balaji, K.; Saxena, N.; Behera, N.R.; Kumar, M.K.; Prasad, H.K.; Gedamkar, P.R. (2024). Improved fraud detection in banking systems through machine learning and big data analytics with management key components. Proceedings of the International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 1–6. http://doi.org/10.1109/ACCAI61061.2024.10601803.

[10] Shah, A.; Mehta, A. (2021). Comparative study of machine learning-based classification techniques for credit card fraud detection. Proceedings of the International Conference on Data Analytics for Business and Industry (ICDABI), Sakheer, Bahrain, 53-59. http://doi.org/10.1109/ICDABI53623.2021.9655848.

[11] Aladakatti, D.; G.P.; Kodipalli, A.; Kamal, S. (2022). Fraud detection in online payment transactions using machine learning algorithms. Proceedings of the International Conference on Smart and Sustainable Technologies in Energy and Power Sectors (SSTEPS), Mahendragarh, India, 223–228. http://doi.org/10.1109/SSTEPS57475.2022.00063.

[12] Lochan, S.; Sumanth, H.V.; Kodipalli, A.; Rohini, B.R.; Rao, T.; Pushpalatha, V. (2023). Online payment fraud detection using machine learning. Proceedings of the International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA), Bengaluru, India, 389–394. http://doi.org/10.1109/CIISCA59740.2023.00080.

[13] Abbasi, M.; Shah, M.A. (2022). Credit card fraud detection using machine learning classifiers in stacking ensemble technique. Proceedings of the Competitive Advantage in the Digital Economy (CADE 2022), Hybrid Conference, Venice, Italy, 76–81. http://doi.org/10.1049/icp.2022.2044.

[14] Ravinder, B.; Seeni, S.K.; Prabhu, V.S.; Asha, P.; Maniraj, S.P.; Srinivasan, C. (2024). Web data mining with organized contents using Naïve Bayes algorithm. Proceedings of the 2nd International Conference on Computer, Communication and Control (IC4), 1-6.

[15] Han, X.; Zhu, X.; Pedrycz, W.; Li, Z. (2023). A three-way classification with fuzzy decision trees. *Applied Soft Computing*, *132*, 109788. https://doi.org/10.1016/j.asoc.2022.109788.

[16] Kurani, A.; Doshi, P.; Vakharia, A.; Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science*, *10*, 183-208. https://doi.org/10.1007/s40745-021-00344-x.

[17] Hu, J.; Szymczak, S. (2023). A review on longitudinal data analysis with random forest. *Briefings in Bioinformatics*, *24*, bbad002. https://doi.org/10.48550/arXiv.2208.04112.

[18] Ding, X.; Feng, C.; Yu, P.; Li, K.; Chen, X. (2023). Gradient boosting decision tree in the prediction of NOx emission of waste incineration. *Energy*, *264*, 126174. https://doi.org/10.1016/j.energy.2022.126174.