ICIC Express Letters
Volume **20**, Number **4**, April **2026**

 ISSN 1881-803X
pp. **455–463**

ENHANCED RANDOM FOREST WITH ENSEMBLE LEARNING FOR FINANCIAL FRAUD DETECTION IN ONLINE MOBILE TRANSACTION

Teguh Wahyono[1,*] and Navida Wahyu Ananda[2]

[1]Faculty of Information Technology
Satya Wacana Christian University
Notohamidjojo Street, Salatiga 50711, Indonesia
*Corresponding author: teguh.wahyono@uksw.edu

[2]Faculty of Economics and Business
Satya Wacana Christian University
Diponegoro Street, Salatiga 50711, Indonesia
232021070@student.uksw.edu

Received May 2025; accepted August 2025

Abstract. *This research is motivated by the increasing prevalence of criminal and fraudulent activities in online financial transactions, highlighting the need for an effective anomaly detection model based on artificial intelligence and machine learning. This study proposes a detection model using the Random Forest algorithm optimized through ensemble learning techniques. Random Forest was selected due to its strong performance in handling large, multi-dimensional datasets with varying scales. The optimization process involves hyperparameter tuning to enhance predictive accuracy and speed, as well as the integration of Extreme Gradient Boosting (XGBoost) to form a hybrid ensemble model. The research methodology follows the CRoss Industry Standard Process for Data Mining (CRISP-DM), a structured approach for developing machine learning models. Experimental results demonstrate that the proposed model achieves a precision of 1.00, recall of 0.94, F1-score of 0.97, and accuracy of 99.97%, outperforming baseline models such as Logistic Regression, Decision Tree, and Naïve Bayes. These results indicate that the proposed approach significantly enhances detection performance for fraudulent activities in digital financial transactions.*

Keywords: Financial fraud detection, Online transaction, Random Forest, Extreme gradient boosting, Ensemble learning

1. **Introduction.** Digitalization has made human life easier in various aspects, including the way society conducts financial transactions. Financial transactions such as payments, withdrawals, transfers or remittances, and even credit or loans can be done online through smartphones or notebooks [1]. Digitalization provides benefits and offers various conveniences. However, it also presents new challenges related to financial fraud. This is evident from the increasing crimes in online financial transactions. In the Global Financial Crime Report, Nasdaq reported that globally in 2023 there will be banking financial fraud with an estimated loss of $485.6 billion [2]. Meanwhile, the Alloy State of Fraud Benchmark Report reported a significant increase in financial fraud incidents in 2024, especially among banks, financial technology companies, and credit unions [3]. According to the report, more than 50% of the increase was in business fraud incidents, and more than 66% of the increase occurred in consumer fraud.

The increasing prevalence of various electronic transaction crimes highlights the importance of developing a fraud detection model in digital financial transactions. Effective fraud detection requires fast and accurate data processing to minimize the risk of losses for

DOI: 10.24507/icicel.20.04.455

companies and the public [4]. Machine learning methods need to be applied to analyzing large and diverse transaction data quickly. Machine learning is an artificial intelligence method that uses algorithms and machine learning to automatically process and analyze data [5]. Machine Learning (ML) has the ability to identify patterns of anomalous events, enabling early detection of potential criminal activities [6].

This research proposes the use of a machine learning algorithm, Random Forest, for detecting financial fraud in online transactions. Random Forest is known for its high accuracy as it employs ensemble learning, combining predictions from multiple decision trees to make the final decision [7]. This helps reduce prediction errors that may occur when using only a single decision tree. Random Forest is also recognized for its ability to handle imbalanced data. In fraud detection cases, legitimate transactions often outnumber fraudulent ones [8,9]. Random Forest excels at handling imbalanced data due to its capacity to utilize different decision trees that focus on various data subsets [10]. Furthermore, to obtain the best model, model optimization should be conducted. This research proposes the optimization of Random Forest, which includes tuning hyperparameters to enhance predictive power and model speed. Another optimization is the implementation of ensemble learning by combining Random Forest with Extreme Gradient Boosting (XGBoost) to improve performance.

The problem-solving approach used is the CRoss Industry Standard Process for Data Mining (CRISP-DM), which is a standard approach for developing machine learning-based models [11]. This method consists of several research stages starting from understanding business needs, understanding data, preparing data, modeling, evaluation, and deployment [12]. Through these stages, this research successfully developed a high-performance model for anomaly detection in online financial transactions, which can effectively assist in identifying and reducing fraudulent activities.

This study contributes to the academic field by developing an improved fraud detection model that enhances the traditional Random Forest algorithm through the integration of ensemble learning techniques, specifically by combining it with Extreme Gradient Boosting (XGBoost). The research also applies systematic hyperparameter tuning to boosting the model's predictive accuracy and computational efficiency, especially in addressing the challenges of imbalanced transaction data. Overall, the resulting model offers a more robust and accurate approach to detecting fraudulent activities in online financial transactions, supporting advancements in digital financial security and machine learning applications. The structure of this paper is arranged as follows. The opening section presents the introduction and outlines the background of the study. The following section describes the research methodology applied. Next, the results obtained and their discussion are provided in the third section. Finally, the paper concludes with key findings and suggestions for future research directions.

2. **Related Research.** Research on fraud detection in financial transactions has been extensively conducted. Some studies have utilized statistical approaches in detecting financial fraud [13,14,18]. Statistical models analyze transactional data to identify outliers or patterns that significantly deviate from expected behaviors. These models can be based on the mean, standard deviation, or other statistical measures [15,16]. While these anomaly detection models are relatively simple and accurate, they have weaknesses when dealing with large datasets [16]. Another approach to anomaly detection involves using machine learning approaches with various methods such as regression [17,18], classification [19,20], clustering [21,22], and neural networks [23].

By employing appropriate statistical methods and machine learning techniques, the system will effectively identify and prevent anomalies in real time [16]. However, previous research results have highlighted various challenges in developing machine learning models, including issues such as data imbalance, evolving fraud techniques, and the presence of

false positives to ensure the accuracy and efficiency of anomaly detection systems [21,23]. Among the various algorithms in machine learning, Random Forest is considered one of the methods capable of addressing several existing issues. In fraud detection cases, imbalanced data often occur, where the number of legitimate transactions far exceeds fraudulent transactions. Random Forest excels at handling imbalanced data due to its ability to use different decision trees that focus on various data subsets [8,9,24]. Random Forest is also known for its robustness against noise in data, which refers to information that is irrelevant or variables that are not related to predictions. The Random Forest algorithm can filter out less important variables and focus on those that truly have an impact.

The Random Forest algorithm is an ensemble learning method used for classification and regression tasks [25]. This algorithm operates by constructing many decision trees during training and combining their results to enhance prediction accuracy and control overfitting [10]. The Random Forest algorithm operates in two main phases, which characterize its architecture and functioning. The first phase is combining a number N of decision trees to create the Random Forest. The second phase is making predictions for each tree created in the first phase and then selecting the best prediction result. The detailed steps can be explained as follows. In the first step, the Random Forest algorithm randomly selects a sample from the provided dataset. It then creates a decision tree for each selected sample and obtains the prediction results from each decision tree created. The next step involves the voting process for each prediction result. For classification problems, the mode (most frequently occurring value) is used, while for regression problems, the mean (average value) is employed. Subsequently, the algorithm selects the most commonly chosen prediction result (the highest vote count) as the final prediction [8-10].

While previous studies have explored fraud detection using statistical models and various machine learning techniques such as classification, clustering, and neural networks, many of these approaches either focused on standalone algorithms or lacked specific optimizations for real-time mobile financial transactions. Furthermore, existing research has not adequately addressed the challenges of data imbalance and model tuning in a combined framework. In contrast, this study proposes an enhanced approach that integrates Random Forest with Extreme Gradient Boosting (XGBoost) through ensemble learning, specifically optimized to improve performance in detecting fraud within imbalanced datasets typical of mobile online transactions. By applying systematic hyperparameter tuning and ensemble strategies, this work aims to offer a more robust, accurate, and scalable solution compared to previous models.

3. **Methodology.**

The method of this research follows the Cross Industry Standard Process for Data Mining (CRISP-DM), which is a standard for developing data science and machine learning-based models. As depicted in Figure 1, the CRISP-DM methodology can be summarized into four main stages of model development [26]. The four steps are as follows: 1) data understanding, focusing on comprehending the data; 2) data preprocessing; 3) modeling; and 4) evaluation and bencharking [11,12].

The first stage is dataset understanding and exploration. This stage involves understanding the data requirements for model development. It includes collecting relevant data, conducting initial data exploration to understand its characteristics, assessing data quality, and identifying potential data issues [12].

The dataset used is the Synthetic Financial Datasets for Fraud Detection provided by Kaggle. This data comprises synthetic financial transaction data generated by the PaySim mobile money simulator. PaySim simulates digital financial transactions based on a sample of real transactions taken from one month's financial records of a digital financial service with over 3000 data points. The original logs were provided by a multinational

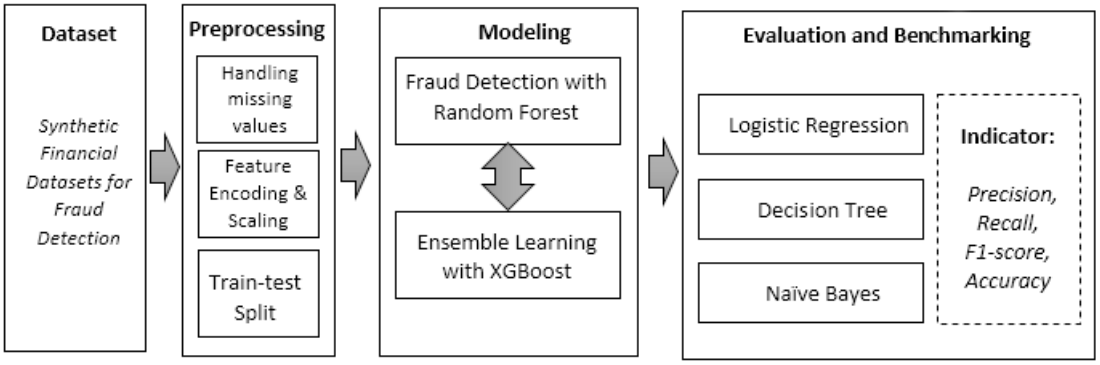

FIGURE 1. Developing a machine learning model with CRISP-DM

company, a mobile financial services provider currently operating in more than 14 countries worldwide. Consequently, this dataset resembles normal transaction operations and includes anomalous behaviors that can be utilized for fraud detection and financial crime prevention. The second stage is data preprocessing. This stage involves cleaning the data by handling missing or invalid values, data transformations such as encoding categorical variables or normalization, and preparing a suitable dataset for analysis needs. The main objective is to prepare the data for analysis in modeling processes.

The next stage is model development. This phase begins by selecting an appropriate model or algorithm for the project, training the model using training data, and validating the model to ensure it performs well [11]. The core of this research lies in developing an enhanced fraud detection model using the Random Forest (RF) algorithm combined with Extreme Gradient Boosting (XGBoost) through ensemble learning. Random Forest is chosen due to its robustness against overfitting and noise, and its ability to handle imbalanced data. The next optimization involves implementing ensemble learning by combining Random Forest with Extreme Gradient Boosting or XGBoost to enhance model performance. XGBoost is a boosting method that utilizes gradient boosting to improve model performance by iteratively correcting errors from previous models. Random Forest, on the other hand, is a bagging method that builds many decision trees in parallel and combines their results to make more stable and accurate predictions [7,8]. Therefore, the combination of these two methods means leveraging the tree structure from XGBoost but using the Random Forest process in feature selection and tree formation, deriving benefits simultaneously from both methods.

Let $D = \{(x_i, y_i)\}_{i=1}^{n}$ be the dataset, where $x_i \in R^d$ and $y_i \in \{0, 1\}$. Random Forest constructs B decision trees $h_1(x), h_2(x), \ldots, h_B(x)$, each trained on a bootstrap sample of data. The final prediction $\hat{y}$ for classification is made by majority voting:

$$\hat{y} = \text{mode}(h_1(x), h_2(x), \ldots, h_B(x)) \tag{1}$$

XGBoost enhances this ensemble by sequentially training weak learners to minimize the regularized loss:

$$L(\phi) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

where l is the loss function, and $\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_j^T w_j^2$ is the regularization term to avoid overfitting. The final hybrid model integrates both algorithms by weighted ensemble or voting schemes, where weights are assigned based on validation performance.

Subsequently, the evaluation stage aims to assess the model using previously unused data (test data), measure the model's performance against pre-defined criteria, and ensure that the model meets business objectives and resolves the defined issues. After the

evaluation and benchmarking stages, the deployment of the best-performing model can be carried out. The trained model is applied to the production environment, integrating the model into existing business processes and developing maintenance and monitoring strategies to ensure sustained model performance.

4. Result and Discussion.

4.1. Dataset exploration.

As explained in the research method section, the dataset for this study was obtained from synthetic financial transaction data generated by the PaySim mobile money simulator. Randomly selected, it consists of 510,793 records and comprises 11 columns or variables. These variables include step, type of transaction, amount, old original balance, new original balance, name of the receiving customer, old destination balance, new destination balance, fraud transaction indicator, and flagged fraud indicator. The dataset handles five transaction types: cash-in, cash-out, debit, payment, and transfer. Fraud transactions in the dataset are transactions conducted by fraudulent agents in the simulator, exhibiting behavior aimed at gaining profit by taking control of customers' account and attempting to take funds by transferring them to another account and then cashing them out of the system. From the total of 510,793 records in the dataset, there are 1,140 fraudulent transactions (0.2% of the total transactions), while valid transactions are 509,653 (99.8% of the total transactions).

4.2. Data preprocessing.

Data preprocessing is a crucial initial step in the machine learning model development. This stage involves various techniques to clean and prepare raw data for use by the machine learning model. In this model development, data preprocessing involves removing duplicate data, checking for missing values, feature encoding, and feature scaling.

The first step in preprocessing is checking for missing values. Handling missing values is a critical step in data preprocessing to ensure data quality and consistency before using it in a machine learning model. If missing values are found, they can be addressed by either removing them or filling in the missing values through interpolation, using methods like mean, median, mode, or predictive values. The next step is to check and remove duplicate data. This check and removal process is done using the command data.drop_duplicates (inplace=True), resulting in the output "there are no duplicates found in the data".

The next step is performing feature encoding. This is a stage in preprocessing that converts categorical data into numeric values that can be utilized by machine learning algorithms. The method used is One-Hot Encoding, which transforms nominal categorical data into Boolean representations, particularly for the transaction type variable. Table 1 displays the results of One-Hot Encoding applied to the transaction type variable in this research dataset.

Table 1. Results of encoding with one-hot encoding

	type_CASH_IN	type_CASH_OUT	type_DEBIT	type_PAYMENT	type_TRANSFER
0	1	0	0	0	0
1	0	1	0	0	0
2	0	1	0	0	0
3	0	0	0	1	0
4	1	0	0	0	0

The next step is performing feature scaling, which involves adjusting the feature values in the dataset to a specific scale. This is crucial in machine learning because many algorithms are sensitive to the scale of the data. Feature scaling also ensures that all features contribute proportionally to the model training. Feature scaling is carried out using MinMaxScaler(), which transforms feature values to be within the range $[0, 1]$. Table 2

Table 2. Results of feature scaling

Type	Amount	Oldbalance original	Newbalance original	Oldbalance destination	Newbalance destination
CASH_IN	0.038099	0.002474	0.012268	0	0
CASH_OUT	0.00643	0	0	0.030969	0.042703
CASH_OUT	0.019289	0.000404	0	0	0.004648
PAYMENT	0.00018	0.000383	0.000337	0	0
CASH_IN	0.016823	0.000527	0.004852	0	0

illustrates the results of feature scaling, where the data has been converted to the $[0, 1]$ range.

The next step is determining which features are the dependent variable (y) or independent variables (X). In this study, the X variables are step, type, amount, oldbalanceOrig, newbalanceOrig, oldbalanceDest, and newbalanceDest. The y variable or dependent variable is the isFraud variable. The final step in this stage is performing a train-test split on the dataset. Train-test split is a technique used to divide the dataset into two subsets, one for training the model (training set) and the other for testing the model (testing set). The method used is stratified sampling, ensuring that the class distribution in the training and testing subsets remains the same as in the original dataset. This is essential to avoid bias in model evaluation, especially when the dataset has imbalanced classes. In this study, the dataset is divided into 80% for training data and 20% for testing data.

4.3. **Fraud detection modeling with enhanced Random Forest.** Based on the background and research objective, modeling was conducted using the Random Forest algorithm with several strategies to enhance model performance, particularly for fraud detection in the financial sector. Performance improvement was achieved through various methods, including hyperparameter tuning and ensemble learning techniques. In the first experiment, tuning was performed on the hyperparameters n_estimators and max_depth in the Random Forest Classifier. Max_depth represents the number of trees (decision trees) to be built in the Random Forest. Therefore, the optimal values for these two hyperparameters need to be found, and the experimentation in this stage discovered that the optimal values for accuracy were n_estimators at 30 and max_depth at 30. These results can be seen in Table 3 below.

Table 3. Results of tuning hyperparameters n_estimators and max_depth

N_estimator	Max_depth	Accuracy score
5	10	0.99941
10	10	0.99947
20	20	0.99957
30	**30**	**0.99958**
50	50	0.99955
100	100	0.99955

A larger number of trees tend to enhance model performance because more trees imply more opinions that can collectively improve predictions. However, after reaching a certain point, adding more trees may not significantly improve accuracy and will only increase computation time and complexity. Similarly, a larger depth value allows trees to learn more details from the data, but if the depth is too large, it can potentially lead to overfitting, especially when trees grow too deep and capture noise in the data.

The next optimization involves implementing ensemble learning by combining Random Forest with Extreme Gradient Boosting or XGBoost to enhance model performance [27].

To implement this ensemble learning, this research utilized the XGBRFClassifier library in Python. Some hyperparameter settings conducted in the experiment involve setting n_estimators = 30, subsample = 0.8, and colsample_bynode = 0.2. The n_estimators parameter is used to determine the number of trees to build. The value 30 was chosen as it is the optimal value from previous experiments. Next, the subsample parameter represents the proportion of samples used to build each tree. This parameter helps reduce overfitting by making each tree more independent of one another. Some references recommend values of 0.8 or 0.9 for the subsample parameter. The colsample_bynode parameter is for the proportion of features used in splitting each node and aims to introduce more variation at each node, similar to Random Forest. With these settings, there was an increase in accuracy to 0.9997. Consequently, the use of ensemble learning has clearly demonstrated an enhancement in the accuracy of the developed model.

4.4. **Evaluation and benchmarking.** The next stage involves benchmarking, which entails comparing the proposed method with several other machine learning algorithms. The performance of the optimized Random Forest algorithm is compared with other algorithms such as Logistic Regression, Decision Tree, and Gaussioan Naïve Bayes. Each model is compared using several indicators, namely Precision, Recall, F1-score, and Accuracy.

Figure 2 shows the summary of model performance obtained using the 'classification_report' command from the scikit-learn library. Furthermore, Table 4 presents a comparative summary of the results from using these four algorithms. From the table, it can be observed that the Random Forest algorithm exhibits the best performance with a precision value of 1, a recall value of 0.94, an F1-score of 0.97, and an accuracy of 0.9997. A precision value of 1 indicates that all instances predicted as positive by the model are indeed positive. Therefore, in this case, there are no false positives or falsely predicted positive data. With a recall value of 0.94, the model successfully identifies 94% of all true positive instances in the dataset. In other words, out of all instances that should be predicted as positive, the model accurately identifies 94% of them.

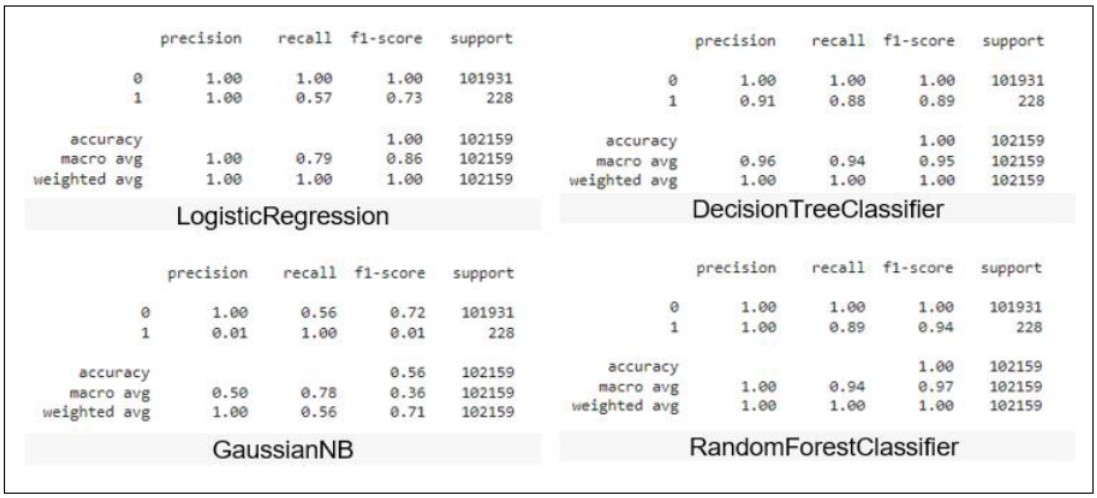

FIGURE 2. Model performance comparison

Furthermore, the F1-score of 0.97 indicates that the model has a good balance of precision and recall, with a very high combined value of both metrics. The F1-score represents the harmonic mean of precision and recall, providing a single measure that considers both metrics. The Random Forest algorithm also has the highest accuracy value compared to the other three algorithms, namely 0.9997. This accuracy score signifies that the model accurately predicts 99.97% of all instances in the dataset. In other words, the

Table 4. Algorithm comparison

Algorithm	Precision*	Recall*	F1-score*	Accuracy score
Logistic Regression	1.00	0.79	0.86	0.9990
Decission Tree	0.96	0.94	0.95	0.9995
Naïve Bayes	0.50	0.70	0.36	0.5580
Enhanced Random Forest	1.00	0.94	0.97	0.9997

*macro average

model only makes errors in 0.03% of the total predictions. Considering these performance measurements, the Random Forest algorithm is suitable for financial fraud detection using machine learning.

5. **Conclusions.** This study demonstrates that the Random Forest algorithm is one of the best algorithms to apply in the context of financial fraud detection using machine learning. Improving the algorithm's performance can be achieved through proper hyperparameter tuning and implementing ensemble learning by combining the algorithm with Extreme Gradient Boosting or XGBoost. Hyperparameter tuning resulted in the best performance with settings at n_estimators = 30 and max_depth = 30, yielding an accuracy score of 0.99958, higher than other tuning options. The implementation of XGBoost also increased the accuracy value to 0.9997. To validate these results, benchmarking was conducted, comparing the proposed method with several other machine learning algorithms such as Logistic Regression, Decision Tree, and Naïve Bayes. In this comparison, the performance of the proposed method still outperformed the others. Further research is needed to expand the comparison of algorithms, especially involving deep learning algorithms known for handling large datasets, such as Recurrent Neural Network (RNN) or Long Short-Term Neural Network (LSTM). Additionally, using different datasets is essential to determine if the proposed method is equally effective with similar datasets.

Acknowledgment. This research was funded by the Directorate Research and Community Services, Satya Wacana Christian through the Research Program for the Improvement of Lecturer Functional Positions with grant number 014/SPK-JAFA/RIK/06/2024, June 11, 2024.

REFERENCES

[1] A. R. Archanti and A. Rohman, Addressing the factors causing financial statement fraud: A systematic literature review and bibliometric analysis, *Journal Eduvest: Journal of Universal Studies*, vol.4, no.6, DOI: 10.59188/eduvest.v4i6.1501, 2024.

[2] Nasdaq, Global financial crime report: Insights at the intersection of financial crime data & real survivor stories, *Nasdaq Online Magazine*, https://www.nasdaq.com/global-financial-crime-report, 2024.

[3] K. J. McAlpin, Financial fraud statistics for banks, fintechs, and credit unions, *Alloy Content Library*, https://www.alloy.com/blog/2024-fraud-stats-for-banks-fintechs-and-credit-unions, 2024.

[4] H. Taherdoost, A review on risk management in information systems: Risk policy, control and fraud detection, *Electronics*, vol.10, no.24, DOI: 10.3390/electronics10243065, 2021.

[5] N. S. A. Polireddi, An effective role of artificial intelligence and machine learning in banking sector, *Measurement: Sensors*, vol.33, 2024.

[6] F. R. Alzaabi and A. Mehmood, A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods, *IEEE Access*, vol.12, pp.30907-30927, 2024.

[7] D. Shah and L. K. Sharma, Credit card fraud detection using decision tree and random forest, *ITM Web of Conferences, EDP Sciences*, vol.53, 02012, 2023.

[8] A. M. Aburbeian and H. I. Ashqar, Credit card fraud detection using enhanced random forest classifier for imbalanced data, *International Conference on Advances in Computing Research*, pp.605-616, 2023.

[9] W. Yundong, A. Zhulev and O. G. Ahmed, Credit card fraud identification using logistic regression and random forest, *Wasit Journal of Computer and Mathematics Science*, vol.2, no.3, pp.1-8, 2023.
[10] A. R. Jena, S. K. Sen, M. Mishra, S. Banerjee, N. Dey and I. Saha, A comparative analysis of financial fraud detection in credit card by decision tree and random forest techniques, *AIP Conference Proceedings*, vol.2876, no.1, 2023.
[11] M. Elkabalawy et al., CRISP-DM-based data-driven approach for building energy prediction utilizing indoor and environmental factors, *Sustainability*, vol.16, no.17, 7249, 2024.
[12] A. M. Shimaoka, R. C. Ferreira and A. Goldman, The evolution of CRISP-DM for data science: Methods, processes and frameworks, *SBC Reviews on Computer Science*, vol.4, no.1, pp.28-43, 2024.
[13] A. Sudjianto, S. Nair, M. Yuan, A. Zhang, D. Kern and F. Cela-Díaz, Statistical methods for fighting financial crimes, *Technometrics*, vol.52, no.1, pp.5-19, DOI: 10.1198/TECH.2010.07032, 2020.
[14] M. Nooribakhsh and M. Mollamotalebi, A review on statistical approaches for anomaly detection in DDoS attacks, *Information Security Journal: A Global Perspective*, vol.29, no.3, pp.118-133, DOI: 10.1080/19393555.2020.1717019, 2020.
[15] Z. Cındık and I. H. Armutlulu, A revision of Altman Z-score model and a comparative analysis of Turkish companies' financial distress prediction, *National Accounting Review*, vol.3, no.2, pp.237-255, DOI: 10.3934/NAR.2021012, 2021.
[16] S. Trilles, S. S. Hammad and D. Iskandaryan, Anomaly detection based on artificial intelligence of things: A systematic literature mapping, *Internet of Things*, vol.25, 101063, 2024.
[17] Y. Sahin and E. Duman, Detecting credit card fraud by ANN and logistic regression, *Proc. of the International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, Turkey, pp.315-319, 2021.
[18] B. Kolukisa et al., An efficient network intrusion detection approach based on logistic regression model and parallel artificial bee colony algorithm, *Computer Standards & Interfaces*, vol.89, 2024.
[19] N. Tripathy et al., Cryptocurrency fraud detection through classification techniques, *International Journal of Electrical and Computer Engineering (IJECE)*, vol.14, no.3, pp.2918-2926, 2024.
[20] K. Sreekala et al., A hybrid Kmeans and ML classification approach for credit card fraud detection, *The 3rd International Conference for Innovation in Technology (INOCON)*, 2024.
[21] N. R. S. Jebaraj, J. Shekhawat and R. Gupta, An overview of clustering algorithms for credit card fraud detection, *International Conference on Optimization Computing and Wireless Communication (ICOCWC)*, 2024.
[22] H. Ahmad et al., 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, 2023.
[23] B. F. Murorunkwere et al., Fraud detection using neural networks: A case study of income tax, *Future Internet*, vol.14, no.6, 168, 2022.
[24] G. Mulugeta, T. Zewotir, A. S. Tegegne, L. H. Juhar and M. B. Muleta, Classification of imbalanced data using machine learning algorithms to predict the risk of renal graft failures in Ethiopia, *BMC Medical Informatics and Decision Making*, vol.23, no.1, 98, 2023.
[25] I. D. Mienye and Y. Sun, A survey of ensemble learning: Concepts, algorithms, applications, and prospects, *IEEE Access*, vol.10, pp.99129-99149, 2022.
[26] A. M. Shimaoka, R. C. Ferreira and A. Goldman, The evolution of CRISP-DM for data science: Methods, processes and frameworks, *SBC Reviews on Computer Science*, vol.4, no.1, pp.28-43, 2024.
[27] O. M'hamdi, S. Takács, G. Palotás, R. Ilahy, L. Helyes and Z. Pék, A comparative analysis of XG-Boost and neural network models for predicting some tomato fruit quality traits from environmental and meteorological data, *Plants*, vol.13, no.5, 746, 2024.