WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM
e-ISSN: 2788-5879 p-ISSN: 2788-5887

Analyzing Machine Learning Algorithms for Cloud-Based Transaction Fraud Detection

Ali Abbas Jasim Alhchaimi[1]*

[1]Department of Computer Science, DhiQar Education Directorate, 64001, DhiQar, Iraq

*Corresponding Author: Ali Abbas Jasim Alhchaimi

DOI: https://doi.org/10.31185/wjcms.253
Received 09 June 2024; Accepted 27 June 2024; Available online 30 June 2024

ABSTRACT: Problem Statement: Cloud services as part of the financial technologies improve the scalability and effectiveness of the services but at the same time expanding the risks of fraud. Due to the increasing complexity of the fraud schemes, the conventional fraud prevention techniques are not sufficient and require efficient, cloud-based solutions. Context: Financial technology cannot be discussed without mentioning cloud services as they provide scalable and efficient ways of conducting transactions. However, this shift also makes these systems vulnerable to fraud, thus raising questions on the security of online financial transactions. Objectives: This paper aims to compare the performance of different machine learning algorithms in the identification of fraud in cloud contexts. The models compare are logistic regression (LR), decision trees (DT), random forest (RF), SVM, and XG Boost (XGB). Within this context, this study will utilize a large dataset in order to determine which of the ML models is most accurate in identifying fraudulent charges, how these various models can be fine-tuned in order to work with high levels of precision and efficiency for real time fraud detection. Results: In the fraud detection, the XGBoost model was the most accurate than other models followed by Random Forest and Decision Trees models. These models outperformed other models in terms of the area under the ROC curve, precision, recall, and the number of false positives while also correctly capturing the patterns of multiple types of fraud in transactions. Conclusion: Ensemble and boosting models such as XGBoost and Random Forest offer a strong mechanism for identifying fraud in cloud-based financial systems. The features that make it easy to handle big data and the flexibility to learn new fraud trends improve the security of transactions in the cloud.

Keywords: Cloud-Security, Fraud-Detection, Fraud-Transaction, ML, Evaluation, Performance.

1. INTRODUCTION

The emergence of cloud computing led to such dramatic changes as enabling the data to be stored, processed, and accessed on the basis of scalable resources and services provided through the Internet. especially in financial industry, now capable of deployment and development of fintech services which are so user-friendly and accessible [1]. Yet this digital transformation has an unwanted side and from it emerges new vulnerabilities and a broader area for malevolent parties to exploit. Among these security challenges, transaction fraud emerges as a significant threat, characterized by unauthorized financial activities aimed at deceitfully diverting funds [2], [3]. Detecting and preventing these fraudulent transactions is paramount to maintaining the integrity, reliability, and trust in financial services provided through cloud platforms. Transaction fraud in cloud environments poses complex challenges due to the sophisticated tactics employed by fraudsters, the massive volume of transactions, and the dynamic nature of cloud services. Traditional fraud detection systems often struggle with high false positive rates, limited scalability, and the inability to adapt to evolving fraud patterns. Moreover, the intrinsic characteristics of cloud computing, such as distributed resources and multi-tenancy, complicate the monitoring and analysis of transactions for fraud detection. There is a critical need for advanced detection systems that can efficiently process large datasets, adapt to new fraud strategies, and minimize the impact on legitimate transactions [4]–[7]. While there is extensive research on fraud detection, there remains a gap in the application and optimization of ML algorithms specifically for detecting transaction fraud in cloud-based financial services. Existing studies have either focused on traditional data centers or have not fully explored the potential of various ML algorithms in the nuanced context of cloud environments [8]. Additionally, there is a lack of comprehensive comparative analysis

*Corresponding author: akbhud719@gmail.com

among different ML algorithms to identify the most effective approach or combination of approaches for this specific application. This research aims to fill the identified gap by conducting an in-depth analysis of various ML algorithms for detecting transaction fraud within cloud-based systems. The study contributions can be summarized as follows:

- Comparative Analysis: We provide a comprehensive comparison of several ML algorithms, including XGBoost, Random Forest, Decision Trees, Support Vector Machines, and Logistic Regression, in terms of their effectiveness in detecting transaction fraud in a cloud environment.
- Enhancement and Deployment: We then demonstrate the ability of such algorithms to handle Big Data for cloud transactional data based on the scalable, precise and efficient performances approach.
- Framework Proposal: Based on our findings, we propose a framework for integrating the most effective ML algorithms into cloud-based financial services to enhance fraud detection capabilities, reduce false positives, and ensure real-time processing.

Our study not only contributes to the academic discourse on cloud security and fraud detection, but also offers practical insights and tools for financial institutions looking to bolster their defenses against transaction fraud in cloud environments.

The reminder of the paper is outlined as follows: Section 2 presents the literature review. Section 3 explains the research methodology. in addition, Section 4 discuss the experimental setup and paper findings. whereas Section 5 concludes this paper.

2. LITERATURE REVIEW

The advent of cloud computing has significantly revolutionized the financial services sector, offering enhanced scalability, efficiency, and accessibility for handling financial transactions [9], [10]. This technological evolution, however, has escalated the complexity and sophistication of fraudulent activities, challenging traditional detection mechanisms. Such technologies as ML and deep learning algorithms with powerful and scalable capabilities of fraud detection are regarded as key technologies that help to solve the issue of fraud taking place in the cloud-based systems. The algorithms able to process and analyze vast datasets in real time form an essential part of rapid adaptations and are used for identifying dynamic identification. As shown in Table 1, a number of seminal works have introduced machine learning and deep learning approaches for credit card fraud detection utilizing cloud environments and real-time capability. The current study is distinctive in the practical implementation of a shopping fraud detection system based on the ensemble of a good variety of machine learning algorithms, i.e., LR, DT, RF, SVC, and especially XGB in the cloud computing context [11], [12]. This approach is underscored by a significant achievement: the XGBoost model, a hybrid deep learning algorithm, can boast of its impressive testing accuracy, of 99.4266%, together with AUC score, 0.998563. This model demonstrates its power as it overcomes the limits of the benchmark criteria of the previous work. It not only surveys but keeps an eye on the intricate, dynamic terrain of the credit card fraud detection with extraordinary, everlasting accuracy [7], [13], [14]. The merger with cloud computing has a special significance as far as both methodology and outcomes of the mentioned study are concerned. Different from some of the previous studies that might look only partially into cloud computing integration or is not predominantly based on real-time computation, the current study focuses the attention on implementing these technological aspects. It covers the application of ML algorithms in cloud computing systems as an indispensable tool for preventing fraudulent activities especially real time appraisal. This alignment is a testament to the burgeoning need among financial institutions and cloud providers to have complex and dynamic counter fraud measures that can quickly adapt to the changing fraud patterns.

Table 1. - Summary of prior studies in the context of ML approaches utilized in fraud detection and analysis.

Article	Methodology	Key Findings	Cloud Integration	Real-Time Processing	Dataset	Performance Metrics	Cloud Challenges Addressed
[15]	ML and DL Algorithms	Achieved high accuracy with CNNs; significant reduction in false negatives	Yes	High	European card benchmark	Accuracy: 99.9%, Precision: 93%	Data scalability, dynamic fraud detection
[16]	ML with GA for Feature Selection	Enhanced detection with optimized feature selection	Yes	Moderate	European cardholders	Not specified	Feature management in cloud environments

[17][18]	Supervised ML Algorithms	Effective differentiation of fraudulent transactions	Partial	Low	Not specified	Comparative analysis of algorithms	Adaptation to evolving fraud patterns
[15]	Random Forest and AdaBoost	Superior detection accuracy and F1-score	Yes	High	Not specified	Accuracy, Precision, Recall, F1-score	Real-time analysis and data processing
[19]	Logistic Regression, Naïve Bayes, KNN	Logistic regression showed better prediction accuracy	No	Low	Not specified	Accuracy, Sensitivity, Specificity	-
[20]	XGBoost with Data Balancing Techniques	XGBoost outperformed other models with data balancing	Yes	High	Imbalanced data	Precision: 0.99, Accuracy: 99%	Handling class imbalance
[21] [22]	Deep Learning Ensemble and SMOTE-ENN	High sensitivity and specificity with deep learning ensemble	Yes	High	Not specified	Sensitivity: 1.000, Specificity: 0.997	Managing class imbalance in cloud
[23]	Intelligent Payment Card Fraud Detection System	Effective predictive models using aggregated features	Partial	Moderate	Public and real transaction records	Enhanced discriminative power with aggregated features	Privacy and data confidentiality
[24]	ML-Based K-means Clustering for Fraud Detection	Improved fraud detection accuracy with K-means clustering	Yes	High	Not specified	Enhanced accuracy and efficiency	Resource allocation and monitoring in cloud

Preventing fraud in real-time is vital in order to prevent as much financial loss as possible and to protect the customer base. However, this process has some drawbacks, mainly because of the increased difficulty in identifying and combating modern fraud schemes, huge data amounts, and the need to provide quick analysis results . Thus, the current literature presents several possibilities and recommendations for tackling these issues:

(i) High Data Volume and Velocity: Real time data processing can be very time consuming hence the need to find ways of processing large amounts of data. Applying the approaches based on the cloud computing and high-performance computing can aid in the effective management of these data streams.

(ii) Sophisticated Fraud Techniques: When fraud schemes are evolving and becoming more subtle, it means that classical detection methods might not be enough. AI and machine learning are most effectively used to identify subtle patterns and to predict fraudulent actions based on the incoming data streams.

(iii) Integration Issues: Another major problem is the analysis of data from different sources and data integrity. The integration of such data is quite complex and the creation of the unified view that can be subsequently used for real-time analysis is only possible with the help of advanced methods.

(iv) False Positives: A concern is the ability to set up fraud detection systems in a way that they do not over detect frauds while at the same time being able to detect frauds. This involves enhancing the algorithms and possibly using techniques that make use of more than one of the models for enhanced detection.

The development of new AI technologies and the rising attention to the cybersecurity threats in the financial sector will greatly enhance the effectiveness of real time fraud prevention. This not only assists in the protection of financial assets but also makes sure that the consumers are in a safer environment while making the transactions.

3. RESEARCH METHODOLOGY

This section details the research methodology conducted in this paper. The methodology includes the dataset description, data preprocessing, and ML model deployment.

3.1 DATASET DESCRIPTION

This study leverages the "Fraudulent Transaction" dataset from the Kaggle Depository, which offers an extensive array of transactional data specifically designed for the identification and analysis of fraud in financial transactions [15]. Key attributes of the dataset include transaction steps (time), types, amounts, and comprehensive account details (originating and destination account identifiers, and balances before and after transactions). Each transaction is distinctly labeled as fraudulent or legitimate, facilitating the deployment of supervised learning models for fraud detection [16]. To discern the linear relationships among the dataset variables, a correlation analysis was executed. This statistical

method quantitatively assesses the extent to which variables move in tandem, employing the Pearson correlation coefficient as a measure as shown in Fig. 1. The coefficients range from -1, indicating a perfect negative correlation, to +1, signifying a perfect positive correlation. A coefficient of 0 denotes the absence of any linear correlation. The analysis of the dataset yielded several noteworthy correlations, as follows: Transaction Step and Amount: The correlation between step and amount is relatively low at 0.022373, indicating a minimal linear relationship between the timing of transactions and their amounts. Account Balances and Fraud Detection: The oldbalanceOrg and newbalanceOrig exhibit an extremely high correlation of 0.998803, implying a nearly perfect linear relationship between the original account's balance before and after a transaction Similarly, oldbalanceDest and newbalanceDest are highly correlated at 0.976569, reflecting a strong linear association in the destination account's balance changes due to transactions. Transaction Amount and Destination Account Balances: The correlation between amount and newbalanceDest stands at 0.459304, a moderate positive correlation suggesting a significant relationship between the transaction amount and the subsequent balance in the recipient's account. Fraud Indicators: The correlation between the amount and isFraud is noted at 0.076688, indicating a slight positive relationship suggesting that larger transactions could be more closely associated with fraud. A critical insight is the correlation between isFraud and isFlaggedFraud, recorded at 0.044109, revealing a weak positive relationship. This suggests that while flagged transactions are more likely to be fraudulent, the flagging mechanism captures only a fraction of fraudulent activity. The quantified correlations highlight the nuanced relationships between transactional features and the occurrence of fraud. The high correlations observed between account balances before and after transactions emphasize the need for models to consider these features carefully, potentially focusing on anomalies that deviate from the expected linear patterns.

FIGURE 1. Heatmap of variables correlations.

The correlation pattern of transaction amounts with changes in destination account /suggest/ reveals that transaction size is an essential feature in devising a fraud detection model. Nevertheless, the findings that the typical correlations

with isFraud taking between 1 and 4% indicate that the fraud detection shouldn't just look for the discrepancies in only one feature and instead they should notice the patterns between a number of features. Though the correlation analysis, quantitative anaylsts from the "Fraudulent Transaction" dataset, show how complicated is the detection of fraud in financial institution. This info discloses that to be effective models predictive firstly are aimed to bypass simple linear relationships and then be equipped with high level of understanding and ability to describe the intricate patterns which are typical for fraudulent activities. The results of this research provide a precedent for understanding fraud detection and the crafting of precise and more dynamic solutions for machine learning algorithms that consider the complex nature of this task.

3.2 DISTRIBUTION OF TRANSACTION TYPES IN THE DATASET

The data set 'Fraudulent Transaction' from Kaggle, which is the basis of this research, covers a wide variety of transaction types including their behavioral and managerial aspects as well as how they are conducted on digital platforms. Typology distribution of transaction types should be used to contextualize the analy-sis and to select the detection models by accounting for features of transaction categories. The dataset comprises a total of 6,262,620 transactions, categorized into five distinct types: CASHOUT, PAYMENT, CASHIN, TRANSFWR, and DEBT. The distribution of these transaction types is as follows: CASH_OUT: it is the most frequent transaction type in the dataset, having 2,237,500 transactions (the largest). A CASH_OUT transaction implies that a certain action is taken to bring money out of the user's account or simply transfer money to another account at risk of fraud due to the direct movement of funds. PAYMENT: The second type of transaction, PAYMENT, occurs most often, registering 2,151,495 transactions. These transactions typically involve direct payments to merchants or individuals for goods and services, highlighting a high volume of everyday financial activity among users. CASH_IN: CASH_IN transactions, totaling 1,399,284, refer to the process of depositing money into an account or reversing a CASH_OUT transaction. This category signifies positive account activity, potentially indicating less direct risk for fraudulent transactions but still requiring vigilance (as illustrated in Fig. 2). TRANSFER: With 532,909 transactions, TRANSFER represents a significant portion of the dataset.

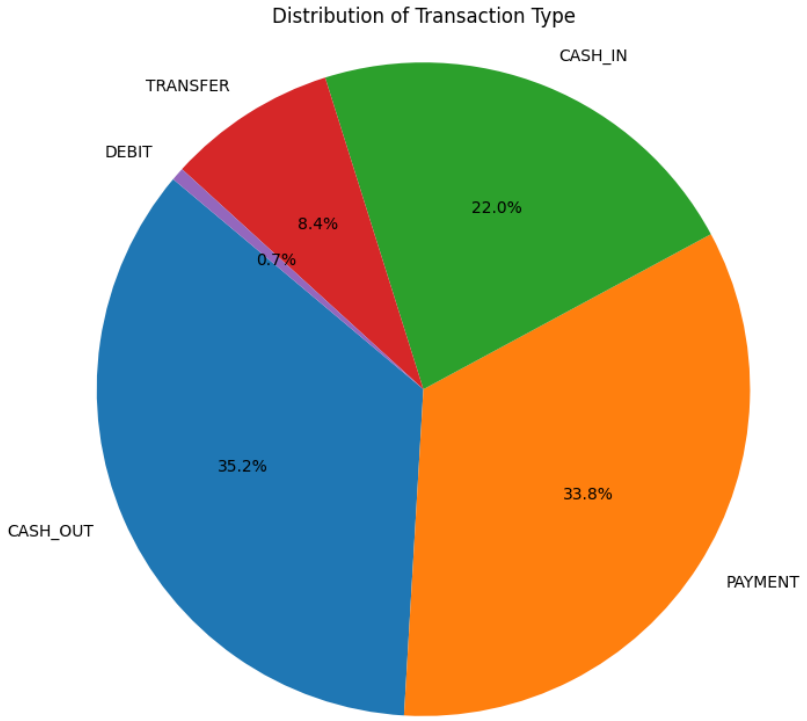


FIGURE 2. Pie illustration to the distribution of transaction Types in the Dataset.

These transactions involve moving funds between user accounts and are of particular interest in fraud detection due to the potential for unauthorized transfers. DEBIT: The least common transaction type, DEBIT, includes 41,432 transactions. DEBIT transactions typically involve withdrawals or payments from an account using a debit card or direct debit instructions, representing the smallest risk category for fraud within the dataset. The distribution of transaction types provides a foundational understanding of the dataset's composition, revealing the prevalence of transactions that directly involve the movement of funds (CASH_OUT and TRANSFER) as well as routine financial activities (PAYMENT, CASH_IN, DEBIT). This distribution has several implications for the development of fraud detection models: Targeted Analysis: The prevalence of CASH_OUT and TRANSFER transactions suggests a need for models to

particularly focus on these types for more effective identification of fraudulent activities. Analyzing the patterns and anomalies within these transaction types could yield significant insights into fraud tactics. Feature Engineering: Understanding the distribution allows for nuanced feature engineering, where models can incorporate transaction-type-specific behaviors and patterns as part of the feature set, potentially improving the accuracy of fraud detection. Risk Assessment: The distribution aids in assessing the risk associated with each transaction type, enabling the development of risk-based models that allocate more resources to monitoring high-risk transactions more closely. The comprehensive analysis of transaction types within the "Fraudulent Transaction" dataset underscores the diverse nature of financial transactions on digital platforms. The data analysis of this study is two-fold. First, the distribution and characteristics of each transaction type are understood. And then, it builds the framework for fraud detection developing the sophisticated, precision models. These models will serve their purpose well by avoiding complications in frauds in digital finance, hence providing security and reliability for any transactions over the internet and digital ecosystem.

3.3 TEMPORAL DISTRIBUTION ANALYSIS OF TRANSACTIONS

Temporal Distribution Analysis of Transactions" subsection investigates temporal distribution of the transactions as well as their frequency using the 'step' feature from the dataset (see Fig. 3). It is this analysis that helps us to understand how transaction activities change, highlight the busy times, and spot any unusual patterns which can suggest there is fraud involved. The histogram plot brings transaction frequencies into the picture using a vertical scale, the frequency, and a horizontal scale, the step. The graphics used in this visual aid enable the identification of transactions that are evenly spread out, any patterns of trends in transaction volumes that are cyclic, related to specific times or events, and any anomalies that significantly differ from the typical pattern, which indicates potential fraud. This initial analysis that set the basis for the further exploration and data processing becomes a fundamental step in the process of feature engineering and model development, especially in fraud detection systems where distinguishing between normal and fraudulent patterns is of the essence.

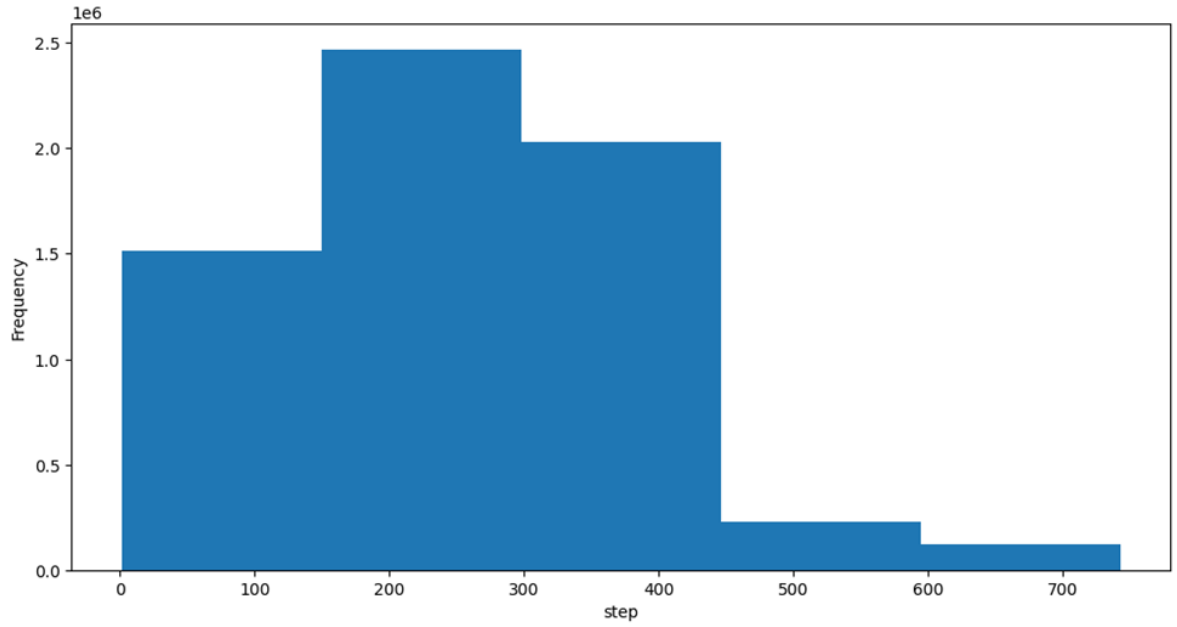


FIGURE 3. Graph illustration to the temporal distribution analysis of transactions types in the dataset.

3.4 CATIGORAL DISTRIBUTION ANALYSIS OF TRANSACTIONS

This section explores the transaction categories in the data set using python labels: "Fraudulent" and "Not Fraudulent" based on the 'isFraud' indicator. Pie chart to clarify the type of fraud and giving a comprehensive view of the fraud commission in the dataset.The process involves: (i) Function Definition: assign_label(x) assigns transaction as "Fraudulent" if x (the 'isFraud' value) is 1, and assigns "Not Fraudulent" to otherwise. (ii) Data Labeling: This label is added to all entries making the 'fraud_label' column to accept 'N' for non-fraud, and 'Y' for fraud. (iii) Visualization: In the pie chart, the revenue earned from fraudulent and non-fraudulent transactions is shown through percents in one decimal place for clear understanding. (iv)Implications: The visualization may be used to provide to an overview of the amount of fraud and the challenge of imbalance, with the number of fraudulent against the non-fraudulent transactions, and this is important in developing effective fraud detection models.

become biased towards the majority class. Strategic Planning: Understanding the proportion of fraudulent transactions assists in prioritizing resources and strategies for fraud detection, guiding the development of more focused and effective analytical models. Communication Tool: Visualizations such as pie charts are effective communication tools that can convey key insights to stakeholders without the need for technical interpretation, facilitating informed decision-making. This section is crucial for providing a foundational understanding of the fraud situation within the dataset, setting the stage for more detailed analytical and modeling efforts aimed at detecting and preventing fraudulent transactions.

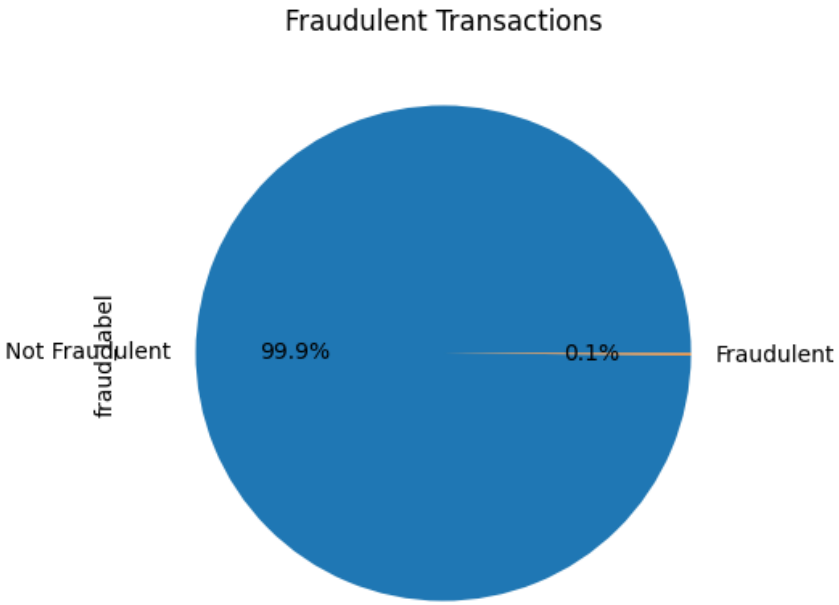


FIGURE 4. Pie to the distribution of fraudulent transactions in the dataset.

3.5 MACHINE LEARNING ALGORITHMS

To address the challenge of detecting fraudulent transactions, we selected five machine learning algorithms based on their proven track record in classification problems and their varied approaches to modeling data. The algorithms include:

• XGB: Gradient boosting machine, an implementation which is scalable and efficient and known for its work with classification tasks.

• RF: It is a collection approach with the creation of various decision trees with the goal of producing an algorithm that is resistant to different types of data and is efficient.

• DT: With a basic layout of a tree leaves model of decisions that prospects would likely take and their outcomes.

• SVM: A robust Kernel SVM that transforms data and then run optimization algorithms to determine an optimal boundary between possible outputs.

• LR: The model incorporates a logistic function to estimate probabilities for binary classifications problems, for which implementation is carried out using a widely known algorithm.

Each algorithm was implemented using the Python programming language, with the help of libraries such as Scikit-learn for model building and training.

3.6 MACHINE LEARNING MODELS TRAINING

Model development included several stages, starting with the division the dataset to the training and testing sets to quantify the performance of each model. Training data was taken 70%, while test data was 30% of the remaining. Each algorithm was tuned by a grid search method to refine hyperparameters and accuracy. In this, the modal value for each parameter was evaluated and the set of parameters that gave the highest accuracy on the validation set was selected. After trial-and-error tuning, each model was trained on the training set. Training comprised of adjusting the model for the data with a goal of pattern recognition of fraudulent and ordinary transactions.

3.7 EVALUATION METRICS

To thoroughly assess the performance of machine learning models in detecting transaction fraud, the study employs a comprehensive set of evaluation metrics. These metrics not only measure the accuracy of the models but also provide insights into their ability to correctly classify transactions as fraudulent or legitimate. Here is a detailed explanation of each metric, along with its mathematical formulation:

Training Accuracy: Training accuracy measures the proportion of correct predictions made by the model on the dataset it was trained on. It is calculated as the number of correct predictions divided by the total number of predictions. Mathematically, it is expressed as:

$$\text{Training Accuracy} = \frac{\text{Number of Correct Predictions on Training Set}}{\text{Total Number of Training Samples}}$$

Testing Accuracy: Testing accuracy measures the proportion of correct predictions made by the model on a new dataset, not seen by the model during training. It serves as an indicator of how well the model generalizes to unseen data. It is calculated as:

$$\text{Testing Accuracy} = \frac{\text{Number of Correct Predictions on Testing Set}}{\text{Total Number of Testing Samples}}$$

Confusion Matrix: The confusion matrix is a table that describes the performance of a classification model on a set of test data for which the true values are known. It outlines the numbers of correct and incorrect predictions broken down by each class. The matrix is structured as follows for a binary classification task:

- True Positives (TP): Correctly predicted positive observations.
- True Negatives (TN): Correctly predicted negative observations.
- False Positives (FP): Incorrectly predicted positive observations (Type I error).
- False Negatives (FN): Incorrectly predicted negative observations (Type II error).

Recall Score: Recall, also known as sensitivity, measures the ability of the model to identify all relevant instances. For fraud detection, it quantifies how many actual fraudulent transactions were correctly identified. It is defined as in formula (1):

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1)$$

-Precision Score

Precision measures the proportion of positive identifications that were actually correct. In the context of fraud detection, it reflects the accuracy of the model in identifying transactions as fraudulent. It is defined as in formula (2):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2)$$

-F1Score

The F1-score measure the harmonic mean of precision and recall, which is an intermediate value between these two factors. Its applications are totally obvious especially when the classes are unbalanced. undefined as in formula (3):

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

-F-BetaScore

The F-beta score is a generalization of the F1 score that can use a beta parameter which allows recall more heavily than precision, or vice versa. A beta with a value > 1 helps recall to be weighed more highly, whereas a beta with a value< 1 helps precision to be weighed more, in comparison. It is defined as in formula (4):

$$\text{F-Beta Score} = (1+\beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}} \quad (4)$$

The combination of these metrics gives a comprehensive report on the models' performance, delivering not just an accuracy assessment but also a measure of its detecting capability regarding window dressing from real ones. These

metrics were chosen to provide a complete picture of every model's performance while not only focusing on accuracy but also minimizing the occurrence of false alarms and misleading signals because that is the essence in the world of against fraud. This approach provides being reliable in the research of comparing performance between machine learning algorithms with bringing to light the detection of fraudulent transactions in cloud-based environments. By mainly focusing on and perfecting these models, and by implementing reliable evaluation metrics, the purpose of this study is to give real-bold insight that will be valuable in improving the secure-ness and authenticity of financial transactions in the cloud.

4. EXPERIMENT SETUP

To assess the performance of the machine learning models for transaction fraud detection, a high-performance environment was used to handle computational and data analysis demands. The hardware used were an Intel Xeon Platinum 8280 processor with a clock speed of 2.7GHz and 28 cores for parallel processing, 256GB DDR4 RAM for in-memory processing of large data, 2TB NVMe SSD for fast data access and model checkpoint storage, and an NVIDIA Tesla V100 GPU with 32GB HBM2 memory for model training. The software environment utilized was Ubuntu 20. 04 LTS for a stable and secured environment, and Python 3. 12 for development, preprocessing, and analysis and several machine learning libraries like scikit-learn, XGBoost, LightGBM, TensorFlow, PyTorch, Panda, NumPy and more. For parallel computation Dask and Joblib were used, and Docker containers were used for maintaining the environment of the experiment.

4.1 EXPERIMENTAL PROCEDURES

The experimental process was carried out in stages, adhering to best practices in machine learning experimentation: Data Preprocessing: The dataset was cleaned, normalized, and split into training and testing sets using automated scripts to ensure consistency across model evaluations. Model Training: Each model was trained separately, utilizing grid search for hyperparameter tuning to find the optimal model configuration. GPU acceleration was implemented to improve the training procedure when it could be critical. Evaluation: Models were judged regarding their performances on the test set where pre-defined set of metrics, such as accuracy, precision, recall, F1 score and F-Beta score, were considered. Data were organized and later on analyzed using an effective approach. Performance Monitoring: HTOP and NVIDIA System Management Interface (nvidia-smi) both tools are used to monitor the utilization of the given computing environment and acknowledge any potential bottlenecking of the experimental process. The establishment of a high-tech computing environment was a turning point in gaining trust in the reliability and accuracy of the experiments' outcomes. It made the work with complex data-sets and frames of algorithms much easier and faster and also guaranteed the absolute reliability and accuracy of the study's outcomes. Future research could explore the scalability of the proposed models and their performance in even larger datasets or more complex fraud detection scenarios, leveraging advancements in computing hardware and software technologies.

4.2 RESULTS AND DISCUSSION

The table presents a comprehensive evaluation of several machine learning models on a dataset presumably aimed at fraud detection (see Fig. 5) and Table 2. The metrics used for evaluation include Training Accuracy, Testing Accuracy, Confusion Matrix, Recall Score, Precision Score, F1 Score, and F-Beta Score. In addition, the performance of each model based on these metrics: This research assessed various machine learning models for fraud detection, emphasizing performance, confusion matrix results, and Area Under the Curve (AUC) values:

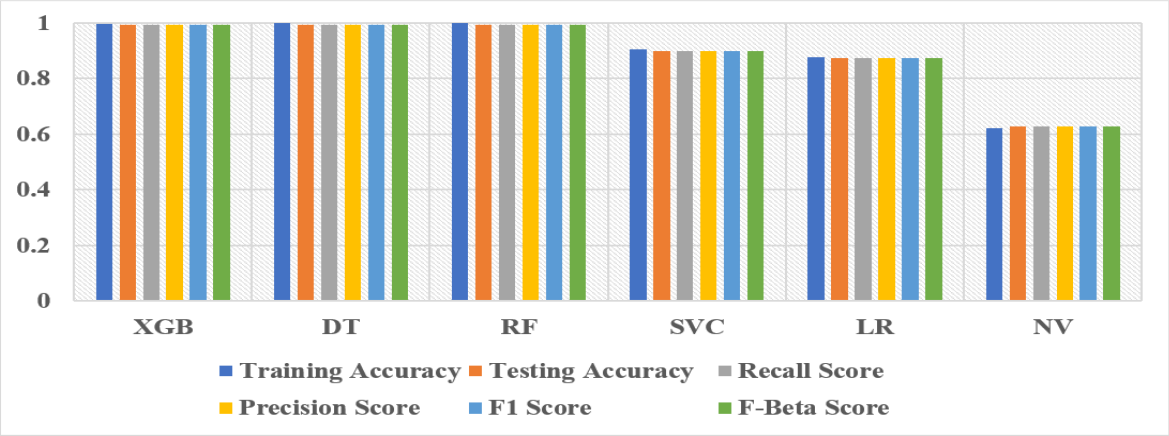
-SVM: Achieved a performance score of 0.899656, with a notable bias towards false positives (173) and minimal false negatives (2), indicating challenges in handling the dataset's complexity.

-LR: Showed modest performance with scores of 0.872133, alongside 95 false positives and 128 false negatives, suggesting difficulties in class distinction.

-NV: Recorded the lowest scores at 0.628440, with a high false positive rate (648), indicating significant misclassification issues.
Top Performers: XGBoost, Decision Trees, and Random Forest excelled, highlighting the advantages
of ensemble and boosting techniques in managing complex data and mitigating overfitting as shown in Fig. 6. Area Under the Curve (AUC) Values:

-XGBoost: Highest AUC of 0.998563, reflecting superior fraud detection capabilities.

FIGURE 5. - A comprehensive comparison among ML utilized models based different performance



metrics

-RF: AUC of 0.997170, benefiting from decision aggregation.

-DT: AUC of 0.993696, showing strong non-linear pattern recognition.
-SVM: AUC of 0.959753, indicating solid performance.
-LR: AUC of 0.949988, demonstrating decent effectiveness.
-NB: Lowest AUC of 0.838057, limited by feature independence assumptions.

The study's methodology involved data preprocessing, model training with hyperparameter tuning, evaluation on accuracy, precision, recall, and efficiency, and performance monitoring to ensure effective computational resource use. Future research directions include scalability testing, deep learning exploration, and enhancing real-time detection.

Table 2. - Results of compassion among different ML models form different performance metrics: Model Training, Accuracy Testing, Accuracy, Con-fusion Matrix, Recall Score, Precision Score, F1-Score, and F-Beta Score.

Rank	Model	Training Accuracy	Testing Accuracy	Recall Score	Precision Score	F1 Score	F-Beta Score
3	XGB	0.996702	0.994266	0.994266	0.994266	0.994266	0.994266
1	DT	1.000000	0.993693	0.993693	0.993693	0.993693	0.993693
2	RF	1.000000	0.993693	0.993693	0.993693	0.993693	0.993693
5	SVM	0.903785	0.899656	0.899656	0.899656	0.899656	0.899656
0	LR	0.876828	0.872133	0.872133	0.872133	0.872133	0.872133
4	NV	0.622311	0.628440	0.628440	0.628440	0.628440	0.628440

Using ensembles and boosting methods (RF and XGB) helps to obtain higher fraud detection, due to the better understanding of the interactions between the variables and the absence of overtraining. Despite this being a strong point of Decision Trees, showing how such relations can be expressed inherently is important. However, even if Logistic Regression is noted to be quite simple, it should be admitted that it produces highly influential results; thus, it would be quite logical to conclude that linear relationships indeed have important roles within this particular field. Based on the results above, low accuracy of naive Bayes is an implication of the fact that it is hard to consider independence of feature while at the same time consider fraud detection. This has been considered to point to the fact that Support Vector Machines were extremely capable of handling high dimensionality and have been observed to be

inarguably a great tool. However, the most important role of experimental studies here is as follows: they allow to select and to improve recipes that are superior to the ones used in art by fraude detection systems, and in the end efficiency will be the driving forc

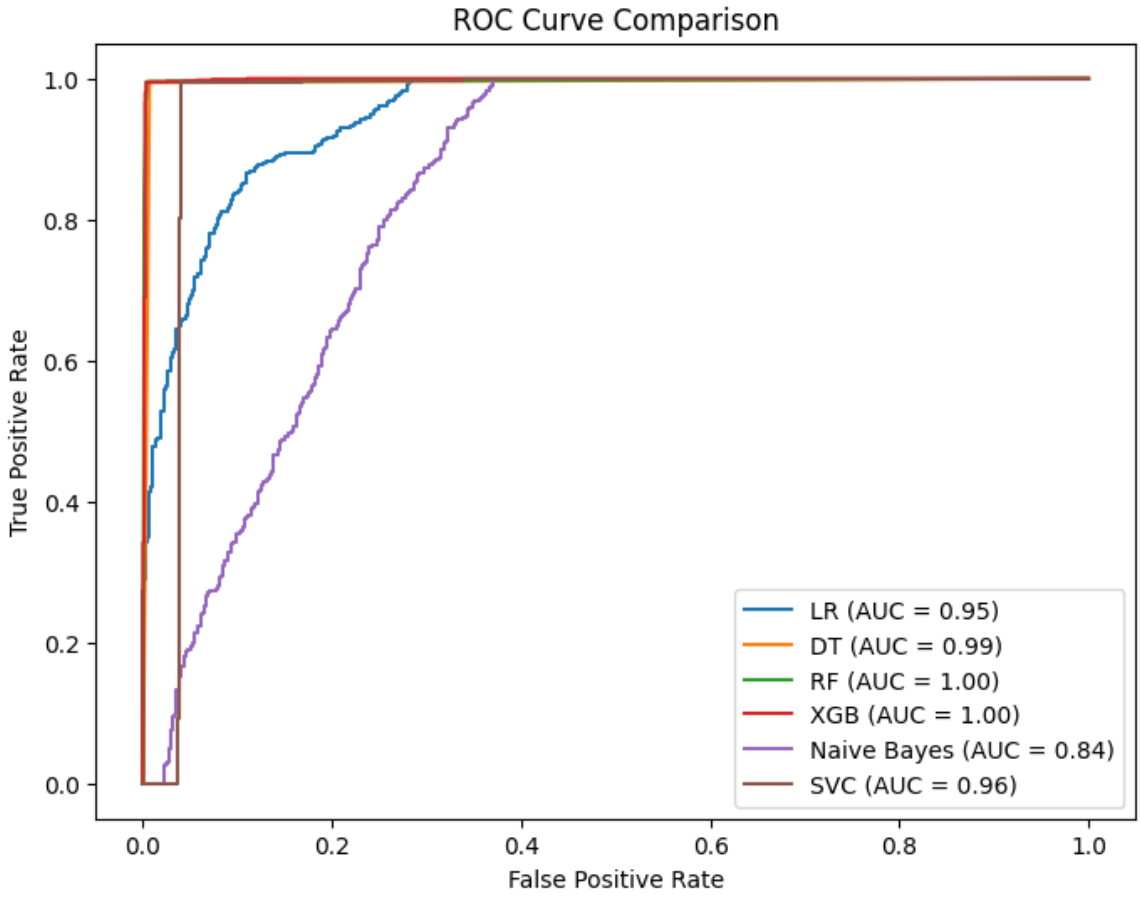


FIGURE 6. - A comprehensive comparison AUC values for utilized ML models

4.3 SCALABILITY AND ADAPTABILITY TO NEW FRAUD TYPES

In the cloud platform, resources can be self-allocated depending on the need at a given time to make the system run smoothly at any time that there is high or low traffic. The deployment of such a structure is important in order to manage the quantities of data that in real-time fraud detection. Flexibility in uncovering new forms of fraud is made possible by the machine learning algorithms which means new data can be learned from. Specifically, authors have pointed out Decision Trees, Random Forests, and XGBoost for high accuracy and continuous improvement of the model's parameters as new instances of fraud are detected. These models are updated with new data from the current environment, and due to this, the details of new fraud schemes are learned and the system grows with threats as they develop. This continuous learning process is backed by the cloud infrastructure as this supplies the adequate computational resources for retraining of models. In addition, applying of ensemble techniques has the positive effect on both the scale and flexibility. Since the models are trained separately, integrating them in the final system enables the system to eliminate the errors that individual models might bring and thereby increase the reliability of the whole system which in a real-time application is very important to users. This means that rather than having to rely on one particular model that the system was trained on, it also allows for the system to be more robust to changes in fraud patterns since possibly different models are more effective at identifying different types of fraud. Thus, it is clear that the development of the proposed system with a cloud

architecture allows it to be elastic, and the utilization of a set of flexible, iterative machine learning algorithms ensures its efficiency against emerging fraud threats.

5. CONCLUSION AND FUTURE SCOPE

In this paper, the effectiveness of different machine learning algorithms like XGBoost, Decision Trees, and Random Forests is analyzed to detect fraud in cloud-based systems. From the result analysis, it is seen that the model with the highest accuracy and the best request time performance is XGBoost. Nonetheless, the following limitations were noted; the data used in the current study was diverse, thus, the models may not be generalizable across different datasets and there are inherent model biases that may affect results. Also, the need for high-performance computing to solve these problems restricts the applicability and adaptability of these approaches in actual implementations.

From the present work several potential future research directions can be identified. First, to improve the fraud detection capacity, there is a possibility to develop the application of deep learning techniques for pattern analysis. This entails using the Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to infer the temporal dependencies and relationships in the data. Secondly, incorporating real time fraud detection mechanisms can greatly enhance the efficiency and timeliness of the fraud prevention measures so that action can be taken at the time of occurrence of the fraud.

Another important direction is to create models that are easily transportable and can be easily deployed on any cloud environment with the least possible challenges. Also, if the models are designed in such a way that they use less computational power, then the more sophisticated methods of fraud detection can be made available to organizations which may be operating on tight budgets. Lastly, it is necessary to increase the interpretability of the machine learning models as it will foster trust and provide the stakeholders with the ability to explain and check the decision-making process taken by the model.

From the managerial perspective, the implications of these findings are far-reaching. From this research, it is possible to identify the strengths and weaknesses of the current approaches to fraud detection to improve the existing systems for the benefit of organizations. The use of high accurate models such as XGBoost can increase the effectiveness and productivity of the fraud detection procedures. However, the managers need to consider the drawbacks, including the requirement of powerful computational infrastructure and the possibility of the models' bias, and manage these constraints effectively. Supporting the development of the larger infrastructure and continuous research on the more efficient and understandable models will guarantee the appropriateness and flexibility of the fraud detection systems in the face of new threats.

Finally, future research can help to develop more sound and efficient approaches to detecting fraud and, therefore, enhance the security of users and organizations functioning in the cloud environment. This will not only reduce the cost of financial losses but will also increase the dependability of cloud-based systems which will, in turn, contribute to the development of long-lasting business strategies and customer confidence.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] L. Hasimi, D. Zavantis, E. Shakshuki, and A. Yasar, "Cloud Computing Security and Deep Learning: An ANN approach," Procedia Comput. Sci., vol. 231, pp. 40–47, 2024.
[2] R. Salama and F. Al-Turjman, "Mobile cloud computing security issues in smart cities," in Computational Intelligence and Blockchain in Complex Systems, Elsevier, 2024, pp. 215–231.
[3] S. H. Z. Al-Enzi, S. Abbas, A. A. Abbood, Y. R. Muhsen, A. A. J. Al-Hchaimi, and Z. Almosawi, "Exploring Research Trends of Metaverse: A Bibliometric Analysis BT - Beyond Reality: Navigating the Power of Metaverse and Its Applications," 2023, pp. 21–34.
[4] A. Mughaid, I. Obeidat, L. Abualigah, S. Alzubi, M. S. Daoud, and H. Migdady, "Intelligent cybersecurity approach for data protection in cloud computing based internet of things," Int. J. Inf. Secur., pp. 1–15, 2024.
[5] P. Rani, S. Singh, and K. Singh, "Cloud computing security: a taxonomy, threat detection and mitigation techniques," Int. J. Comput. Appl., pp. 1–14, 2024.
[6] U. A. Butt, R. Amin, M. Mehmood, H. Aldabbas, M. T. Alharbi, and N. Albaqami, "Cloud security threats and solutions: A survey," Wirel. Pers. Commun., vol. 128, no. 1, pp. 387–413, 2023.

[7] Y. R. Muhsen, N. A. Husin, M. B. Zolkepli, N. Manshor, and A. A. J. Al-Hchaimi, "Evaluation of the Routing Algorithms for NoC-Based MPSoC: A Fuzzy Multi-Criteria Decision-Making Approach," IEEE Access, 2023.
[8] A. A. J. Al-Hchaimi, N. B. Sulaiman, M. A. B. Mustafa, M. N. B. Mohtar, S. L. B. M. Hassan, and Y. R. Muhsen, "Evaluation Approach for Efficient Countermeasure Techniques Against Denial-of-Service Attack on MPSoC-Based IoT Using Multi-Criteria Decision-Making," IEEE Access, vol. 11, pp. 89–106, 2023, doi: 10.1109/ACCESS.2022.3232395.
[9] A. A. J. Al-hchaimi, M. F. Alomari, Y. R. Muhsen, N. Bin Sulaiman, and S. H. Ali, "Explainable Machine Learning for Real-Time Payment Fraud Detection: Building Trustworthy Models to Protect Financial Transactions," in International Conference on Explainable Artificial Intelligence in the Digital Sustainability, 2024, pp. 1–25.
[10] Y. R. Muhsen and A. A. J. Al-hchaimi, "Modelling Intelligent Agriculture Decision Support Tools to Boost Sustainable Digitalization: Evidence from MCDM Methods," in International Conference on Explainable Artificial Intelligence in the Digital Sustainability, 2024, pp. 93–105.
[11] A. G. Wadday, A. A. J. Al-hchaimi, and A. J. Ibrahim, "IOT Energy Consumption Based on PSO-shortest Path Techniques," Recent Adv. Electr. Electron. Eng. (Formerly Recent Patents Electr. Electron. Eng., vol. 13, no. 7, pp. 993–1000, 2020.
[12] A. A. J. Al-Hchaimi, N. Bin Sulaiman, M. A. Bin Mustafa, M. N. Bin Mohtar, S. L. B. Mohd Hassan, and Y. R. Muhsen, "A comprehensive evaluation approach for efficient countermeasure techniques against timing side-channel attack on MPSoC-based IoT using multi-criteria decision-making methods," Egypt. Informatics J., vol. 24, no. 2, pp. 351–364, 2023, doi: https://doi.org/10.1016/j.eij.2023.05.005.
[13] N. A. Husin, M. B. Zolkepli, N. Manshor, A. A. J. Al-Hchaimi, and A. S. Albahri, "Routing Techniques in Network-On-Chip Based Multiprocessor-System-on-Chip for IOT: A Systematic Review," Iraqi J. Comput. Sci. Math., vol. 5, no. 1, pp. 181–204, 2024.
[14] A. A. J. Al-Hchaimi, W. N. Flayyih, F. Hashim, M. S. Rusli, and F. Z. Rokhani, "Review of 3D Networks-On-Chip Simulators and Plugins," in 2021 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), 2021, pp. 17–20. doi: 10.1109/PrimeAsia51450.2021.9701472.
[15] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," IEEE Access, vol. 10, pp. 39700–39715, 2022.
[16] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," J. Big Data, vol. 9, no. 1, p. 24, 2022.
[17] S. Khatri, A. Arora, and A. P. Agrawal, "Supervised machine learning algorithms for credit card fraud detection: a comparison," in 2020 10th international conference on cloud computing, data science & engineering (confluence), 2020, pp. 680–683.
[18] Y. R. Muhsen, N. A. Husin, M. B. Zolkepli, N. Manshor, A. A. J. Al-Hchaimi, and H. M. Ridha, "Enhancing NoC-based MPSoC Performance: A Predictive Approach with ANN and Guaranteed Convergence Arithmetic Optimization Algorithm," IEEE Access, 2023.
[19] F. Itoo, Meenakshi, and S. Singh, "Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection," Int. J. Inf. Technol., vol. 13, no. 4, pp. 1503–1511, 2021.
[20] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam, "Unbalanced credit card fraud detection data: a machine learning-oriented comparative study of balancing techniques," Procedia Comput. Sci., vol. 218, pp. 2575–2584, 2023.
[21] T. R. Noviandy, G. M. Idroes, A. Maulana, I. Hardi, E. S. Ringga, and R. Idroes, "Credit Card Fraud Detection for Contemporary Financial Management Using XGBoost-Driven Machine Learning and Data Augmentation Techniques," Indatu J. Manag. Account., vol. 1, no. 1, pp. 29–35, 2023.
[22] I. D. Mienye and Y. Sun, "A deep learning ensemble with data resampling for credit card fraud detection," IEEE Access, vol. 11, pp. 30628–30638, 2023.
[23] M. Seera, C. P. Lim, A. Kumar, L. Dhamotharan, and K. H. Tan, "An intelligent payment card fraud detection system," Ann. Oper. Res., vol. 334, no. 1, pp. 445–467, 2024.
[24] Z. Huang, H. Zheng, C. Li, and C. Che, "Application of Machine Learning-Based K-Means Clustering for Financial Fraud Detection," Acad. J. Sci. Technol., vol. 10, no. 1, pp. 33–39, 2024.