ISSN: 2643-671X

Research Article

Journal of Agriculture and Horticulture Research

Using Gradient Boosting Machines (GBM) Algorithm to Enhance Mobile Money System Security

Joshua Mortey*

Kwame Nkrumah University of Science and Technology- Kumasi Ghana

***Corresponding Author**
Joshua Mortey, Kwame Nkrumah University of Science and Technology- Kumasi Ghana.

Submitted: 2025, Sep 27; **Accepted:** 2025, Oct 26; **Published:** 2026, Jan 16

Citation: Mortey, J. (2026). Using Gradient Boosting Machines (GBM) Algorithm to Enhance Mobile Money System Security. *J Agri Horti Res, 9*(1), 01-09.

Abstract

This study proposes a machine learning framework to enhance mobile money security through fraud detection. Motivated by increasing threats like fraud and unauthorized access, it evaluates Gradient Boosting Machine (GBM) algorithms XGBoost, LightGBM, CatBoost, and AdaBoost using a public mobile transaction dataset. After preprocessing, training, and testing, model performance is assessed using accuracy, precision, recall, F1-score, and AUC. Results show GBM models outperform traditional methods, with XGBoost and CatBoost achieving about 99% detection accuracy and high precision. These models demonstrate strong potential for real-time fraud detection, enhancing financial security, inclusion, and user trust. The study also suggests further optimization for resource-limited environments and evolving cyber threats.

Keywords: Mobile Money, System Security, Machine Learning

1. Introduction

Mobile money systems have revolutionized financial transactions in developing regions by providing accessible and convenient banking services through mobile devices. This innovation has significantly advanced financial inclusion, enabling millions of low-income and rural users to participate in the formal economy [1]. According to Global Mobile Money Adoption Survey, user numbers increased from 45 million in 2009 to a projected 360 million by 2012, reflecting its transformative impact [2]. Despite these benefits, the rapid growth of mobile money has also led to security challenges, including fraud, unauthorized access, and money laundering, as seen in platforms like MTN and Telecel in Ghana [3]. Traditional security mechanisms have proven inadequate in countering evolving cyber threats due to their limited adaptability and static nature. This shortcoming has encouraged the adoption of advanced machine learning techniques to detect and prevent fraudulent activities. Among these, Gradient Boosting Machines (GBM) have emerged as a particularly effective solution because of their strong predictive capabilities and resistance to overfitting [4]. By combining multiple weak learners typically decision trees into a powerful ensemble, GBM efficiently identifies non-linear patterns and subtle irregularities in transactional data, making it suitable for real-time fraud detection in mobile money systems.

The increasing sophistication of cyberattacks, including phishing, SIM swapping, and data breaches, exposes critical vulnerabilities such as weak authentication systems and poor network security [5-7]. further highlight the limitations of existing security infrastructures. While regulatory frameworks, such as those established by FATF in 2020, provide essential guidelines, their implementation largely depends on the technological capabilities of service providers. Hence, the integration of machine learning, particularly GBM, presents a scalable and proactive approach to combating financial fraud. This study aims to strengthen mobile money security by developing a GBM-based fraud detection framework capable of identifying anomalies in real time. Its objectives include assessing the effectiveness of existing security measures, designing a GBM model for fraud detection, and comparing its performance with traditional models like decision

trees and logistic regression. Through this approach, the study seeks to enhance fraud detection accuracy, reduce false positives, and improve overall system resilience. Ultimately, implementing GBM in mobile money platforms will not only safeguard financial data but also foster greater user trust and sustainable growth in digital financial ecosystems.

1.1. Overview of Mobile Money System

Mobile money systems have transformed global financial landscapes since their emergence in the early 2000s, particularly in regions lacking traditional banking infrastructure. The launch of M-Pesa in Kenya in 2007 marked a revolutionary shift [8]. Operated by Safaricom, M-Pesa enabled users to deposit, withdraw, and transfer funds, as well as make payments directly from their mobile phones. Its success was driven by the widespread use of mobile devices, limited access to conventional banking, and the need for affordable financial services, especially in rural areas. M-Pesa demonstrated how mobile money could bridge financial gaps, promoting inclusion and stimulating economic development by integrating unbanked populations into formal financial systems [9]. Following Kenya's success, countries across Africa, Asia, and Latin America replicated and adapted the model to their social and economic contexts.

In developing economies, mobile money has been especially transformative, enabling users to conduct transactions without bank accounts and overcoming barriers such as low income, geographic isolation, and lack of identification. According to GSMA, by the end of 2021, over 1.2 billion mobile money accounts existed globally. This growth stems from three main factors: the widespread availability of mobile phones, improved network coverage, and supportive regulatory policies [10]. The COVID-19 pandemic further accelerated adoption as lockdowns and distancing measures increased reliance on digital financial services. Many first-time users turned to mobile money for safe and convenient transactions, underscoring its critical role in promoting financial inclusion and resilience worldwide.

Regional Classification	Registered Accounts	Active Accounts	Transaction Volume	Transaction Value (USD)
Global	1.6 billion	401 million	65 billion	1.26 trillion
East Asia and the Pacific	361 million	68 million	8 billion	180 billion
Europe and Central Asia	22 million	5 million	345 million	6 billion
Latin America and the Caribbean	57 million	22 million	1 billion	35 billion
Middle East and North Africa	59 million	6 million	357 million	21 billion
South Asia	336 million	82 million	10 billion	185 billion
Sub-Saharan Africa	763 million	218 million	45 billion	832 billion

Table 1: Regional Distribution of Mobile Money Accounts and Transactions

Regional Classification	Registered Accounts	Active Accounts	Transaction Volume	Transaction Value (USD)
West Africa	290 million	76 million	12 billion	227 billion
North Africa	18 million	1 million	97 million	4.7 billion
East Africa	390 million	115 million	28 billion	491.8 billion
Central Africa	65 million	22 million	3.7 billion	57.6 billion
Southern Africa	18 million	5 million	490 million	5.3 billion

Table 2: Regional Distribution of Mobile Money in Africa

2. Related Works

The rise of mobile money systems has revolutionized financial transactions but also increased security risks such as fraud and unauthorized access [1]. Machine learning algorithms now provide advanced solutions for detecting and preventing fraudulent activities in these platforms. A research work by uses machine learning classifiers to predict fraud in mobile money transfers [11]. Using real-time transactions, the logistic regression model showed reasonable performance, but the random forest classifier showed outstanding performance. The amount of money transferred was the top feature for predicting money laundering transactions. The author proposed that future research is needed to enhance the logistic regression model and explore the random forest classifier for law enforcement and financial institutions. Highlight that the rising reliance on smartphones for daily activities exposes personal information to risks associated with malware attacks [12]. To address this issue, they propose a two-layer machine learning detection model that utilizes Ensemble Learning and Stacked Generalization techniques to effectively predict and classify the increasing number of attacks targeting Android smartphones. Their model demonstrates superior performance compared to previous research conducted on the CIC-Maldroid-2020 dataset, achieving an impressive accuracy of 99.49% in classifying various types of attacks.

This study explores mobile payments and transaction fraud, focusing on bank card enrollment. Traditional measures like rule-based expert systems and SMS user authentication are insufficient. The authors propose using the improved Gradient Boosting Decision Tree algorithm, XGBoost library, for real-world application. The authors aim to develop a new fraud detection system for a Chinese payment processor [16]. Offered valuable insights into fraud prediction in mobile money transfers through their research on Case-Based Reasoning (CBR), which serves as an alternative to conventional machine learning methods [13]. Their approach incorporated machine learning techniques, such as k-nearest Neighbor (k-NN), to assign weights to parameters within the dataset. Additionally, they employed a genetic algorithm (GA) to optimize the significance of these weights. The authors demonstrated that classifying log information into five distinct contexts and then recombining them into a single dimension rather than treating amount, time, or feature dimensions as separate parameters enhanced the performance of their weighted CBR system, achieving accuracies of 97% and 98%.

Highlighted the importance of data visualization in fraud detection. It uses the PAYSIM dataset as a case study, analyzing 6,362,620 records [14]. The study found that visualization can identify early indications of incompatibility and highlight key findings. This approach can improve the accuracy and efficiency of fraud detection systems, protecting users from financial losses. The paper concludes that data visualization should be integral to any data analysis project, especially in fraud detection [15]. carried out further research on credit card fraud detection utilizing machine learning (ML) algorithms. A different study evaluated five algorithms Random Forest, Naïve Bayes, K-Nearest Neighbor, Logistic Regression, and Multilayer Perceptron on a European dataset comprising fraudulent and legitimate transactions. The results indicated that Random Forest achieved the highest accuracy rate of 99.7%. However, this study had some limitations, as it did not consider hyperparameters, ensemble methods, or other ML algorithms. To overcome these limitations, conducted a comparative analysis of various ML detection techniques for credit card fraud, incorporating hyperparameters and ensemble methods [2]. The findings confirmed that Random Forest remained the top-performing algorithm, with an accuracy rate of 99.77%, followed by Support Vector Machine (SVM), Artificial Neural Network (ANN), Decision Tree (DT), and Logistic Regression (LR).

In another study by employed the same supervised machine learning algorithms logistic regression, random forest, and decision trees it was found that the random forest classifier, when enhanced with the boosting technique, surpassed the performance of both logistic regression and decision trees [17]. This approach achieved an accuracy of 96%, accompanied by an area under the curve (AUC) value of 98.9%. Conducted experiments using naive Bayesian and C4.5 decision tree classifiers to combat fraud in credit card transactions [18]. The effectiveness of these experiments was assessed based on precision, recall, and the area under the precision-recall curve (PRC). The researcher concluded that the C4.5 decision tree algorithm achieved a success rate of 92.74%. offered valuable insights into predicting fraud in mobile money transfers [19]. Their research utilized Case-Based Reasoning (CBR), presenting an alternative to conventional machine learning methods. However, their approach incorporated machine learning techniques, such as k-nearest Neighbor (k-NN), to assign parameter weights within the dataset. Additionally, they employed a genetic algorithm (GA) to optimize the significance levels of these weights. The authors demonstrated that classifying log information into five contexts and then recombining them into a single dimension rather than using the traditional method of treating amount, time, or feature dimensions as separate parameters enhanced the performance of their weighted CBR system, achieving accuracies of 97% and 98%.

This paper employs eight different algorithms to compare and identify the most effective one through comprehensive experimentation. However, current fraud detection methods often face challenges, leading to either false positives or false negatives, which can result in financial losses for both cardholders and financial institutions [20]. In light of this, the present study aims to evaluate the performance of eight machine learning (ML) algorithms in detecting credit card fraud (CCF) using two distinct datasets. The research thoroughly investigates Logistic Regression (LR), Decision Trees (DT), Random Forests (RF), Multilayer Perceptron (MLP), Naive Bayes (NB), XGBoost, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) algorithms to identify the model that yields the most accurate predictions for both datasets. Additionally, to enhance the accuracy and effectiveness of CCF detection, this study incorporates Principal Component Analysis (PCA) for dimensionality reduction, ensuring that key data characteristics are retained and thereby improving the overall detection process [21]. Throughout the research, various performance metrics, including accuracy, precision, F1-score, and recall, will be calculated and analyzed for each model. The results will be visually presented, compared with other models and research efforts, and evaluated based on their accuracy, precision, F1-score, and recall performance. To address the risks of overfitting and underfitting, the paper will utilize cross-validation, a well-established technique in data analysis.

3. Methodology

In this section, the researcher follows a systematic approach to creating Boosting Machines to learn from a set of financial datasets to predict fraudulent acts similar to unseen data. This approach will help to improve the security of the mobile money system. The proposed methodology will encompass several essential phases. The steps involved in the study involve seven (7) steps:

- Data collection
- Data preprocessing
- Exploratory data analyses
- Obtain Cleaned dataset
- Split cleaned dataset into training and testing set
- Model development (fit model)
- Testing, metric evaluation, and prediction

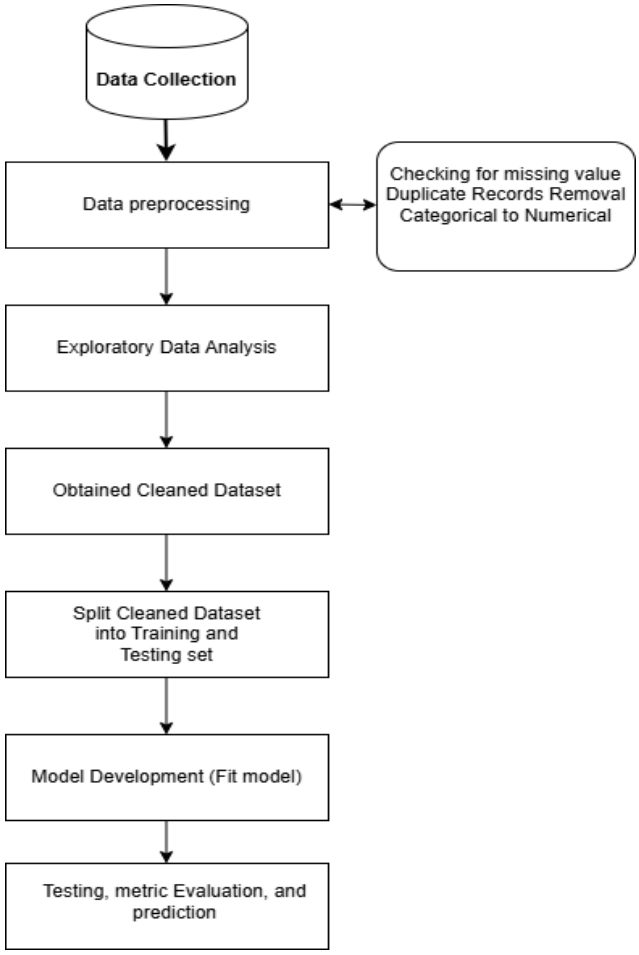

Figure 1: Proposed Methodology

Figure 1. Illustrates the methodology used in this paper. The data was sourced from Kaggle, a publicly accessible repository. The dataset undergoes cleaning, selection, transformation, and visualization to make it suitable for analysis, uncovering 25 characteristics not immediately visible through observation. After refinement, the data is split into training and testing sets. The training set was used to develop predictive models with four Gradient Boosting Machines (GBM) algorithms. Once trained, the models can predict outcomes for new, unseen data. The test set is then used to evaluate the model's performance, determining its predictive accuracy.

3.1. Technical Tools Used

This study utilizes several technical tools to achieve its research objectives. Python serves as the primary programming language, supported by various Python libraries for data collection, preprocessing, analysis, modeling, and visualization. The research environment was configured using Anaconda, with Jupyter Notebook employed for interactive scripting and visualization. All implementations were performed on a Windows operating system using a machine equipped with a 64-bit quad-core processor and 16GB of RAM.

A. Data Collection

The study source data from https://www.kaggle.com/ntnu-testimon/paysim1. This dataset contains mobile money transactions that were generated with the PaySim simulator. The simulation was based on a sample of real transactions gathered by a company that is the provider of the mobile financial service which is currently running in more than 14 countries all around the world. The data is a set of one-month financial logs from a mobile money service implemented in an African country. This dataset was first used by E. A. Lopez-Rojas, A. Elmir, and S. Axelsson in 2016 research and they had successful research. The dataset consisted of 12 feature attributes and 1,048,576 records. It was chosen for its comprehensive inclusion of all attributes relevant to the study's research objectives. These attributes included Payment_type, Trans_amount, customer ID, Old_balance, Current_balance, nameDest, false promotion, and oldbalanceDest. The dataset comprised instances of financial transaction records, with two target variables identifying transactions as either fraud or not fraud.

B. Data Preprocessing

Data preprocessing is a vital phase that ensures raw data is refined for accurate analysis and model development. Using Python and

the Pandas library, the dataset was imported, examined, and cleaned to correct missing or inconsistent values. Addressing missing data was essential to prevent bias and enhance model reliability. The dataset included three categorical features payment type, nameOrig, and nameDest whose unique values were analyzed to understand their impact on the modeling process.

C. SMOTe-ENN for Imbalance Data

The cleaned dataset used in this study displayed a significant class imbalance, with fraud to no fraud ratio of 1% (fraud) to 99% (no fraud). A common strategy for addressing imbalanced datasets is to upsample the minority class. Upsampling can be achieved through various techniques, with one popular method being the Synthetic Minority Oversampling Technique (SMOTE) combined with the Edited Nearest Neighbor (ENN) algorithm. The SMOTE-ENN approach integrates these two methods to enhance the representation of the minority class.

D. Split Cleaned Dataset into Training and Testing

To evaluate the model's performance, the data was split into training and testing sets. The training set accounted for 70% of the data, while the remaining 30% was designated as the validation set. This division ensured that both fraudulent and non-fraudulent instances were included in each set. By organizing the data in this way, the model could be trained and tested on separate samples. The training set comprised 629,145 observations, whereas the test set included 419,430 observations. Furthermore, five-fold cross-validation was employed to conduct a more thorough data analysis.

E. Model Development (Fit Model)

Model development or "fitting the model" is to build a predictive model by applying a chosen algorithm and training it on a given dataset. Four GBM algorithms will be selected for this research, specifically chosen for their proven effectiveness in predicting financial crimes. The algorithms are XGBoost, LightGBM, Catboost, and AdaBoost. The algorithms have been tested extensively in the field and have had a lot of success with financial crime research.

F. Hyperparameter Tuning

Python's Scikit-learn library offers a range of machine-learning algorithms that can be applied to datasets to develop models. It operates using a black-box approach, executing complex computational processes in the background. Parameters were adjusted to improve the model's performance. Tuning means finding the right settings for the algorithm to make it work as well as possible. This helps avoid problems like overfitting, where the model learns too much from the training data and doesn't perform well on new data. There are two main areas where adjustments were made: the training data used to teach the model and the internal settings of the machine learning algorithm that help it process the data.

G. Performance Metric Measurement

At this stage, the algorithms were evaluated for efficiency and predictive performance using a designated test dataset. Each trained model was applied to the test data to assess its accuracy in predicting outcomes, thereby measuring its effectiveness. The final validation involved comparing the model's predictions with actual test results to confirm its reliability and generalization capability.

- **Precision**

Precision measures the proportion of correctly predicted positive instances among all predicted positives, indicating the model's accuracy in identifying true positive outcomes, as represented in equation (1).

$$\text{Precision (P)} = \frac{TP}{TP+FP} \text{ ---------- (1)}$$

- **Recall**

Recall, also known as sensitivity, measures the proportion of actual positives correctly identified by the model, reflecting its ability to capture all relevant positive cases, as shown in equation (2).

$$Recall(R) = \frac{TP}{(TP+FN)} \text{ ---------- (2)}$$

- **F1 Score**

The F1 Score, calculated as the harmonic mean of precision and recall (equation 3), provides a balanced measure of a model's overall performance by combining both accuracy and sensitivity into a single metric.

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

- **Accuracy**

Accuracy measures the overall correctness of a model by calculating the ratio of correctly predicted instances to the total dataset, as shown in equation (4), indicating the model's general reliability and effectiveness.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \text{ ---------- (4)}$$

4. Experimental Evaluation

The experimental framework employs Anaconda version 6.5.4 and Jupyter Notebook to create a flexible and interactive Python-based environment for implementing the Gradient Boosting Machines (GBM) algorithm aimed at enhancing mobile money system security. Anaconda provides a well-managed ecosystem that ensures compatibility and efficient dependency control across all modules. The experiment was conducted on a Windows 11 desktop with 1TB of storage and 8GB of RAM, offering sufficient computational resources. Python's extensive library ecosystem, featuring prebuilt functions and analytical tools, supports effective model development, enabling robust analysis and reliable performance evaluation.

4.1. Summary Statistics

	step	amount	bal_before_transaction	bal_after_transaction	bal_of_recepient_before_transaction	bal_of_receipient_after_transaction	fraud_transaction
count	$1048575	$1048575	$1048575	$1048575	$1048575	$1048575	$1048575
mean	$26	$158666	$874009	$893808	$978160	$1114197	$0
std	$15	$264940	$2971750	$3008271	$2296780	$2416593	$0
min	$1	$0	$0	$0	$0	$0	$0
25%	$15	$12149	$0	$0	$0	$0	$0
50%	$20	$76343	$16002	$0	$126377	$218260	$0
75%	$39	$213761	$136642	$174599	$915923	$1149807	$0
max	$95	$10000000	$38900000	$38900000	$42100000	$42200000	$1

Figure 2: Summary Statistics from the Cleaned Dataset

The dataset analyzed consists of 1,048,575 financial transactions distributed across 95 time steps, encompassing details on transaction amounts, account balances before and after transactions, and a binary indicator for fraudulent activity. The average transaction amount is approximately $158,666, with values ranging from $1 to $10,000,000, indicating substantial variability in transaction sizes. Sender balances before transactions average around $874,009, with a median of $16,002, while recipient balances before transactions average $978,160, with a median of $126,377. Both sender and recipient balances display highly skewed distributions, as 25% of accounts begin with zero balance, whereas a few holds multimillion-dollar amounts, reaching as high as $421 million before and $422 million after transactions. Only a single fraudulent transaction is recorded in the dataset, representing less than 0.0001% of all transactions. This transaction corresponds to the largest recipient balance observed, suggesting it is a significant outlier. Overall, the dataset reflects extreme imbalance and heterogeneity in both account balances and transaction magnitudes, making it particularly suitable for research in financial anomaly detection and fraud identification.

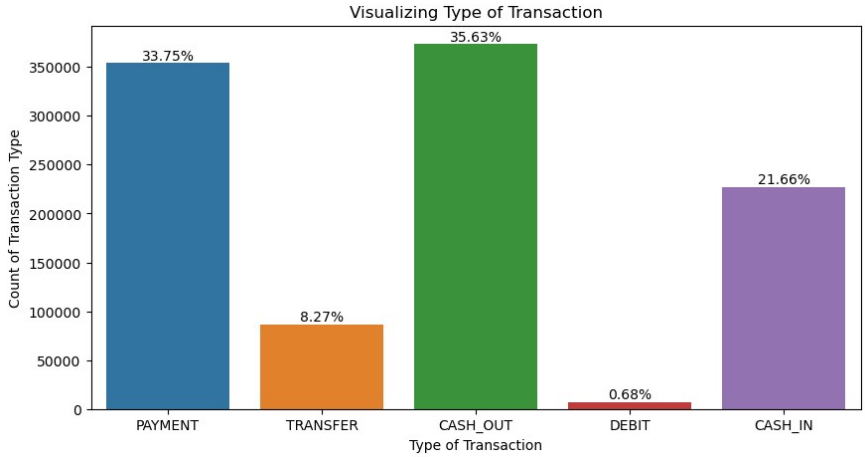

Figure 3: Distribution Different Type of Transaction

As illustrated in Figure 3, the dataset contains five distinct types of transactions: cash-out, cash-in, payment, transfer, and debit. Cash-out transactions are the most common in money laundering schemes, representing nearly 36% of all transactions. Following cash-out are payment transactions (34%), cash-in (22%), transfer (8%), and debit (1%). The majority of transactions (70%) involve cash, with the rest consisting of payments, transfers, and debit card transactions. On average, $159,000 is laundered per transaction, with a standard deviation of $265,000. These results suggest that while money laundering often involves large sums, the amount laundered per transaction can vary significantly across different regions.

4.2. Parameters Employed by Each Model

Algorithm	Parameter	Description / Typical Value
XGBoost	n_estimators	Number of boosting rounds (200)
	learning_rate	Step size shrinkage to prevent overfitting (0.01)

	max_depth	Maximum depth of trees (5)
	subsample	Fraction of samples used per tree (0.6)
	colsample_bytree	Fraction of features used per tree (0.6)
	gamma	Minimum loss reduction for a split (0–3)
	objective	Learning objective (binary:logistic)
LightGBM	n_estimators	Number of boosting iterations (100)
	learning_rate	Controls model update rate (0.01)
	num_leaves	Controls model complexity (120)
	max_depth	Maximum tree depth (5)
	feature_fraction	Fraction of features used per iteration (0.7)
	bagging_fraction	Fraction of data used per iteration (0.8)
	objective	Defines task type (binary)
CatBoost	iterations	Number of boosting rounds (100)
	learning_rate	Step size for weight updates (0.01)
	depth	Depth of trees (4)
	l2_leaf_reg	L2 regularization term (3)
	border_count	Number of splits for numerical features (36)
	loss_function	Objective function (CrossEntropy)
AdaBoost	n_estimators	Number of weak learners (50)
	learning_rate	Weight applied to each classifier (0.01)
	base_estimator	Weak learner (typically DecisionTreeClassifier)

Table 4

4.3. Comparison of Proposed GBM Models Using Performance Metrics Aggregated Scores

Algorithm/model	Precision /Sensitivity	Specificity	Roc Score	Accuracy
XGBoost	99%	81%	99.7%	99.9%
LightGBM	19%	42%	69%	99.7%
Catboost	98%	84%	99.6%	99.9%
AdaBoost	97%	56%	99%	99.9%

Table 5

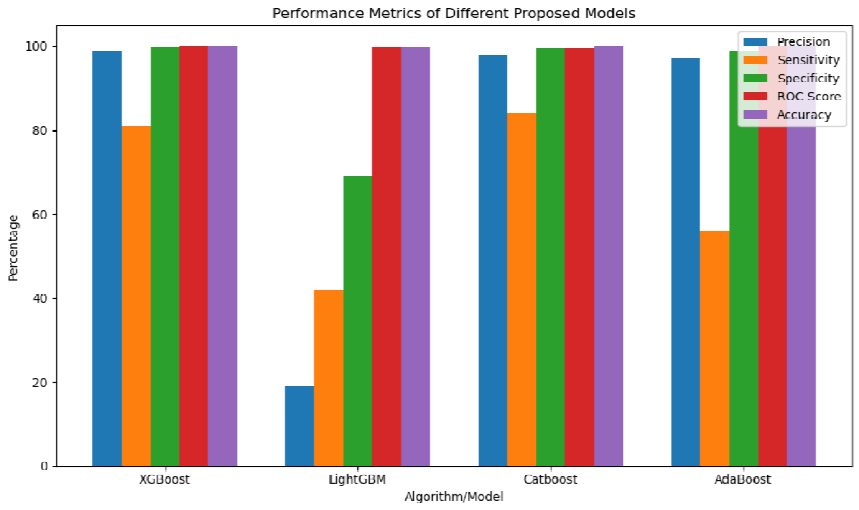

Figure 4: Comparison of Proposed GBM Models

Algorithm/model	Roc Curve (%)	Ranking
XGBoost	99.66%	1st
LightGBM	68.86%	4th
CatBoost	99.57%	2nd
AdaBoost	99.00%	3rd

Table 6

The ROC curve results indicate that XGBoost achieved the highest performance with a score of 99.66%, ranking first among all models. CatBoost closely followed with 99.57%, while AdaBoost recorded 99.00%, showing strong predictive capability. LightGBM, however, performed considerably lower with 68.86%, ranking fourth, suggesting reduced discriminative power compared to the other ensemble models.

5. Conclusion and Findings

This study implemented Gradient Boosting Machine (GBM) algorithms XGBoost, LightGBM, CatBoost, and AdaBoost to enhance the security of mobile money systems through intelligent fraud detection. Using the publicly available Financial Fraud Detection Dataset from Kaggle, data preprocessing was conducted to remove invalid and non-numeric entries, resulting in a refined dataset comprising 1,048,576 records and 11 features. The data was split into 70% for training and 30% for testing using the Scikit-learn library in Python. Each algorithm was trained on the training subset and evaluated using the test subset to measure predictive accuracy and reliability.

Experimental results demonstrated that XGBoost achieved the highest fraud detection rate of 99%, followed by CatBoost (98%), AdaBoost (97%), and LightGBM (lowest performance). In terms of specificity, CatBoost recorded the best true negative rate (84%), followed by XGBoost (81%), AdaBoost (56%), and LightGBM (42%). The ROC curve analysis further supported these findings, with XGBoost attaining 99.7%, CatBoost 99.6%, AdaBoost 99%, and LightGBM 69%. Overall accuracy results showed that XGBoost, CatBoost, and AdaBoost outperformed LightGBM, achieving 99.9% accuracy compared to LightGBM's 99.7%.

The findings affirm that GBM-based algorithms particularly XGBoost, CatBoost, and AdaBoost are highly effective for detecting fraudulent transactions in mobile money systems. This research contributes to financial cybersecurity by demonstrating how machine learning can strengthen fraud detection mechanisms, thereby safeguarding digital financial transactions. The study highlights the increasing relevance of mobile money platforms in developing economies and underscores the importance of data-driven approaches to maintaining financial security and user trust. Future research should address dataset imbalance, which can bias model predictions, by employing data-balancing techniques such as SMOTE. Additionally, subsequent studies may explore hybrid or deep learning models to further enhance fraud detection capabilities and validate model performance in real-world mobile payment environments.

References

1. Abubakar, M., & Sualihu, A. (2025). Financial Inclusion through Fintech Innovations in Ghana: Opportunities, Challenges, and Developmental Impacts. *Challenges, and Developmental Impacts* (June 30, 2025).
2. Koi-Akrofi, J. (2007). Mobile money adoption in Africa: a literature-based analysis. *growth, 7*.
3. Osabutey, E. L., & Jackson, T. (2024). Mobile money and financial inclusion in Africa: Emerging themes, challenges and policy implications. *Technological Forecasting and Social Change, 202*, 123339.
4. Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics, 7*, 21.
5. Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports, 7*, 8176-8186.
6. Ayeboafo, B., Anomah, S., & Amofah, K. (2025). Leveraging blockchain technology adoption in the fight against corruption: An evaluation of Ghana's readiness. *Journal of Economic Criminology, 8*, 100158.
7. Mazele, O., & Amoah, C. (2022). The causes of poor infrastructure management and maintenance in South African municipalities. *Property Management, 40*(2), 192-206.
8. Grzybowski, L., Lindlacher, V., & Mothobi, O. (2023). Mobile money and financial inclusion in Sub-Saharan Africa. *Information Economics and Policy, 65*, 101064.
9. Ahmad, A. H., Green, C., & Jiang, F. (2020). Mobile money, financial inclusion and development: A review with reference to African experience. *Journal of economic surveys, 34*(4), 753-792.
10. Shaikh, A. A., Glavee-Geo, R., Karjaluoto, H., & Hinson, R. E. (2023). Mobile money as a driver of digital financial inclusion. *Technological Forecasting and Social Change, 186,* 122158.
11. Lokanan, M. E. (2023). Predicting mobile money transaction fraud using machine learning algorithms. *Applied AI Letters, 4*(2), e85.
12. Shafin, S. S., Ahmed, M. M., Pranto, M. A., & Chowdhury, A. (2021, December). Detection of android malware using tree-based ensemble stacking model. In *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)* (pp. 1-6). IEEE.
13. Adedoyin, A., Kapetanakis, S., Samakovitis, G., & Petridis, M. (2017, November). Predicting fraud in mobile money transfer using case-based reasoning. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence* (pp. 325-337). Cham: Springer International

Publishing.
14. Al-Sayyed, R., Alhenawi, E. A., Alazzam, H., Wrikat, A. A., & Suleiman, D. (2024). Mobile money fraud detection using data analysis and visualization techniques. *Multimedia Tools and Applications, 83*(6), 17093-17108.
15. Ahmed, M. H. (2023). A Review: Credit Card Fraud Detection in Banks using Machine Learning Algorithms. *ScienceOpen Preprints*.
16. Chakaravarthi , S., Vaishnnave, M. P., & Jagadeesh , M. (2023). A novel light GBMoptimized long short-term memory for enhancing quality and security in web service recommendation system..
17. Dhankhad, S., Mohammed, E., & Far, B. (2018, July). Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In *2018 IEEE international conference on information reuse and integration (IRI)* (pp. 122-125). IEEE.
18. Husejinovic, A. (2020). Credit card fraud detection using naive Bayesian and c4. 5 decision tree classifiers. Husejinovic, A.(2020). *Credit card fraud detection using naive Bayesian and C, 4*, 1-5.
19. Adedoyin, A., Kapetanakis, S., Samakovitis, G., & Petridis, M. (2017, November). Predicting fraud in mobile money transfer using case-based reasoning. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence* (pp. 325-337). Cham: Springer International Publishing.
20. Dechow, P. M., Myers, L. A., & Shakespeare, C. (2010). Fair value accounting and gains from asset securitizations: A convenient earnings management tool with compensation side-benefits. *Journal of accounting and economics, 49*(1-2), 2-25.
21. Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., ... & Eshun, J. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal, 6*, 100163.