



A Machine Learning Framework for Mobile Money Fraud Detection Using Gaussian Naïve Bayes: A Transaction-Level Evaluation on the Synthetic Paysim Dataset

Davies Isobo Nelson¹, Abiodun Ibrahim Akintunde²

¹Department of Computer Science, Rivers State University (RSU), Nigeria

²Department of Computer Science, Ignatius Ajuru University of Education (IAUE), Rivers State, Nigeria

ARTICLE INFO	ABSTRACT
Published Online: 06 October 2026	In Mobile money services are increasingly targeted by fraud, including SIM-related abuse such as SIM swapping, which undermines user trust, exposes sensitive financial data, and costs telecom operators and financial providers billions of dollars annually. Traditional rule-based detection systems have proven insufficient for identifying evolving fraud patterns, motivating the adoption of machine learning approaches. This study presents the design and implementation of a transaction-level mobile-money fraud screening system built around the Gaussian Naïve Bayes (GNB) classifier and evaluated on the synthetic PaySim dataset. The system was implemented following the Rapid Application Development (RAD) methodology using object-oriented design techniques. The dataset was split 80%/20% into training and held-out test sets; normalization was fitted on the training data, and the Synthetic Minority Oversampling Technique (SMOTE, sampling ratio 0.1) was applied to the training set only, leaving the test set at its natural class distribution. Furthermore, predicted fraud probabilities were converted to class labels using a decision threshold of 0.3 instead of the conventional 0.5. The developed model achieved 99.91% accuracy, 89.22% precision, 88.67% recall, 88.94% F1-score, 99.95% specificity, an AUC-ROC of 0.9425, and a false positive rate of 0.05%. The findings indicate that Gaussian Naïve Bayes provides an interpretable and computationally practical baseline for transaction-level fraud screening. However, because PaySim contains no SIM-specific signals such as SIM-swap events, device/IMSI changes, or OTP request patterns, the developed model should be understood as a general mobile-money transaction fraud screen rather than a dedicated SIM-fraud detector; the synthetic nature of the dataset and the conditional-independence assumption of Naïve Bayes further limit direct generalization to live telecommunications environments. Future work should incorporate real-world telecom and mobile-money data with genuine SIM-level features, adaptive fraud behaviour, deep learning or sequential models, and real-time stream processing.
Corresponding Author: Davies Isobo Nelson	
KEYWORDS: Machine Learning; Naïve Bayes; Gaussian Naïve Bayes algorithm; Mobile Money Fraud Detection; Transaction-Level Fraud Screening; SMOTE; PaySim Dataset	

I. INTRODUCTION

The global telecommunications landscape has undergone a dramatic transformation over the last two decades. With over 5.4 billion mobile subscribers worldwide as of 2025, mobile phones have evolved beyond simple communication devices into powerful financial instruments, particularly in developing economies where mobile money services have bridged the gap created by limited access to traditional

banking infrastructure [1]. Across sub-Saharan Africa, Southeast Asia, and Latin America, platforms such as M-Pesa, Airtel Money, and MTN Mobile Money have enabled billions of previously unbanked individuals to participate in formal financial ecosystems, sending and receiving money, paying bills, and accessing microloans all through a Subscriber Identity Module (SIM) card.

However, this remarkable democratization of financial services has come with a significant dark side due to the explosive growth of SIM-related fraud. A SIM card is the small chip that authenticates a mobile user on a network [2]. Nonetheless, in the context of mobile money fraud, it serves as both the identity credential and the transaction gateway [3]. When fraudsters gain control of a SIM through techniques such as SIM swapping, SIM cloning, and fraudulent SIM registration, they effectively assume the victim's financial identity [4]. They can intercept One-Time Passwords (OTPs), authorize transfers, drain mobile wallets, and even access linked bank accounts, often before the legitimate user becomes aware [5].

According to the Communications Fraud Control Association (CFCA), telecom fraud alone costs the industry approximately \$38.95 billion annually, with SIM-related fraud representing one of the fastest-growing categories. SIM swap fraud, in particular, has been implicated in high-profile cases involving cryptocurrency theft, corporate espionage, and large-scale financial theft [6]. In countries with high mobile money penetration such as Nigeria, Ghana, Kenya, and South Africa network operators and regulators have reported steep increases in subscriber complaints related to unauthorized SIM activity. In addition, the Nigerian Communications Commission (NCC), for instance, has issued multiple advisories warning subscribers about SIM swap attacks and fraudulent SIM registrations [7].

However, there is a high level of difficulty in detecting these types of SIM-related fraud. Unlike physical theft, which leaves tangible evidence, SIM fraud exploits legitimate channels such as a customer walks into a service center with fake documentation, or a fraudster exploits insider access at a telecom agent [4]. Most of the time, anomalous transactions are often flagged by conventional rule-based systems when significant financial damage has already occurred. A typical traditional Rule-based system operates on predefined thresholds and static conditions. This makes them fragile against evolving fraud strategies.

Fraudsters are adaptive; they can study detection systems and modify their behaviour to evade detection by staying just below transaction thresholds or mimicking normal user behaviour. Nevertheless, despite the growing awareness and increased investment in telecom security, SIM fraud continues to escalate in both frequency and sophistication. The inadequacy of current detection systems can be attributed to the core problem such as the extreme difficulty in handling uncertainty and dynamic fraud patterns by intelligent SIM-specific fraud detection systems. However, in recent years, Computer Science and Cybersecurity researchers have identified Machine Learning algorithms as a compelling solution for fraud detection. Unlike the rule-based systems, ML models learn patterns directly from historical transaction data and can generalize to detect previously unseen fraud strategies [8]. Among the various ML algorithms available, Naïve Bayes classifier holds a

unique position due to how it is probabilistically grounded, computationally efficient, interpretable, and has consistently demonstrated competitive performance in classification tasks involving uncertain or noisy data. Its probabilistic nature makes it particularly well-suited for handling the inherent uncertainty in fraud detection, where the same transaction pattern might be legitimate in one context and fraudulent in another [9].

This study develops a Machine Learning Framework for Mobile Money Fraud Detection using Gaussian Naïve Bayes algorithm. Although SIM-related fraud motivates the work, the scope of the study is transaction-level fraud screening: the publicly available PaySim dataset used for training and testing contains no SIM-specific attributes (such as SIM-swap events, device or IMSI changes, or OTP request patterns), so the system does not detect SIM-specific attacks directly. The system was developed using the Python programming language and an SQLite database, and was trained and tested on the synthetic PaySim mobile-money transaction dataset.

The contributions of the study are includes an end-to-end, interpretable Gaussian Naïve Bayes fraud-screening application, a leakage-free evaluation protocol in which SMOTE is applied to the training data only, and a fully specified SMOTE configuration and decision threshold to support reproducibility.

II. RELATED WORKS

The article of [10] introduces a scenario-labeled synthetic dataset known as “FraudX SimS” for benchmarking anomaly detection in payment transactions, especially fraud detection. The dataset keeps the real-world imbalance between legitimate and fraudulent activity and includes well-defined spatial, temporal, and behavioral features that work well with explainable AI methods. The study evaluate multiple machine-learning baselines using standard performance metrics such as accuracy, precision, recall, F1-score, confusion matrices, AUC-ROC, and AUC-PR, with an emphasis on recall to avoid missing fraud. The study employed SHAP to show how features drive predictions, helping with clearer error analysis and model improvements. However, while synthetic, the dataset is designed for reproducible research and fair comparisons across studies.

[11] proposed a multi-layer fraud detection framework for mobile money transactions by combining Multi-Factor Authentication (MFA) with machine learning. The system employed a three-stage architecture (prevention, real-time fraud detection, and intelligent decision-making). The study utilized the PaySim dataset by testing the performance of Logistic Regression, Random Forest, and Gradient Boosting. The experimental results indicated that Random Forest performs best by achieving 99.95% accuracy, 96.77% precision, 93.18% recall, and an F1-score of 94.94%. The design also includes a five-layer workflow and is validated through functional testing across six key modules.

“A Machine Learning Framework for Mobile Money Fraud Detection Using Gaussian Naïve Bayes: A Transaction-Level Evaluation on the Synthetic Paysim Dataset”

The study of [12] investigated SIM swap fraud and proposed a behavioural analytics framework that monitored user activity patterns including call patterns, SMS frequency, and data usage to detect sudden behavioural shifts indicative of a SIM swap event. The framework used K-Means clustering to establish normal user behaviour profiles and flagged deviations that exceeded two standard deviations from the profile centroid. The system achieved a recall of 86% for SIM swap detection in a dataset provided by a South African telecom operator. The study's major contribution is its specific focus on SIM swap as a fraud category. However, it did not incorporate transaction-level financial data, which limits its applicability to mobile money fraud detection.

[13] compared the performance of Naïve Bayes, Random Forest, and Decision Tree classifiers for credit card fraud detection using a publicly available credit card transaction dataset. The Naïve Bayes classifier achieved an accuracy of 89.3% and an F1-score of 0.87. The authors highlighted the algorithm's speed and interpretability as key advantages, particularly for real-time applications. The author also noted that Gaussian Naïve Bayes was more effective than Multinomial Naïve Bayes for continuous transaction features. This finding directly informs the algorithmic choice in the present study. The limitation of this work is its focus on credit card fraud, which differs structurally from SIM fraud in terms of feature composition and fraud mechanisms.

[14] applied Long Short-Term Memory (LSTM) neural networks to detect SIM-related fraud patterns in a Nigerian telecom operator's dataset. The their proposed LSTM model leverages the temporal nature of transaction sequences, thereby modelling the order and timing of events to detect suspicious patterns that most static models might miss. Their study achieved an accuracy of 94.7% and a recall of 91.3% for SIM fraud detection.

The study of [15] proposed a machine-learning based anomaly detection framework for fraud detection. The workflow includes preprocessing, feature selection, dimensionality reduction, and classification, and was trained on 6.2 million transactions from the PaySim dataset using a 70%/15%/15% split for training, validation, and testing. The authors evaluate six classifiers such as Random Forest, Decision Tree, SVM, RBF-SVM, Logistic Regression, and Gradient Boosting. The reported accuracy for TRANSFER transactions is: 0.9999 (Random Forest), 0.9998 (Decision Tree), 0.9528 (SVM), 0.9635 (RBF-SVM), 0.9673 (Logistic Regression), 0.9977 (Gradient Boosting). For CASHOUT transactions, the accuracy values are: 0.9793 (Random Forest), 0.9937 (Decision Tree), 0.9673 (SVM), 0.9698 (RBF-SVM), 0.9766 (Logistic Regression), 0.9985 (Gradient Boosting). Nevertheless, framework delivers high accuracy across both transaction types, supporting its robustness for fraud detection in large-scale mobile payment systems.

[16] proposed an adaptive framework which captures the co-evolution of fraud strategies and their detection methods. The study created a multi-agent environment with an

adaptive fraudster and a cost-sensitive defense using the PaySim mobile money dataset. Nonetheless, while the defender modifies the evaluation criteria to balance fraud capture and consumer friction, the fraud agent uses an adaptive heuristic inspired by reinforcement learning to dynamically change transaction amounts based on defensive pressure. Their results was processed over 6.3 million transactions, and was reported that while the defender is able to limit the scale of fraud, the attacker learns to evade detection over time, causing fraud recall to deteriorate from 60.38% to 46.74% over ten simulation episodes. In addition, the findings demonstrated a significant trade-off between detection effectiveness and false-positive rates, as more aggressive detection policies resulted in increased customer friction. The authors therefore emphasize the importance of incorporating adaptive attacker behaviour into evaluation environments to develop more resilient and robust financial fraud detection systems

III. METHODOLOGY AND DESIGN

This study adopts the Rapid Application Development (RAD) methodology as its software development framework. The proposed Mobile Money Fraud Detection System is designed using the techniques of Naïve Bayes machine learning (ML) classifier. Here, the system ingests transaction data, preprocesses it through a pipeline of normalization and feature engineering steps, passes it to the trained Naïve Bayes model, and returns a fraud probability score and binary classification for each transaction.

Furthermore, the study adopts client-server architecture, in which the client side comprises the web-based user interface and the server side encompasses the application logic, machine learning module, and database. The high-level architecture is presented in Figure 1.

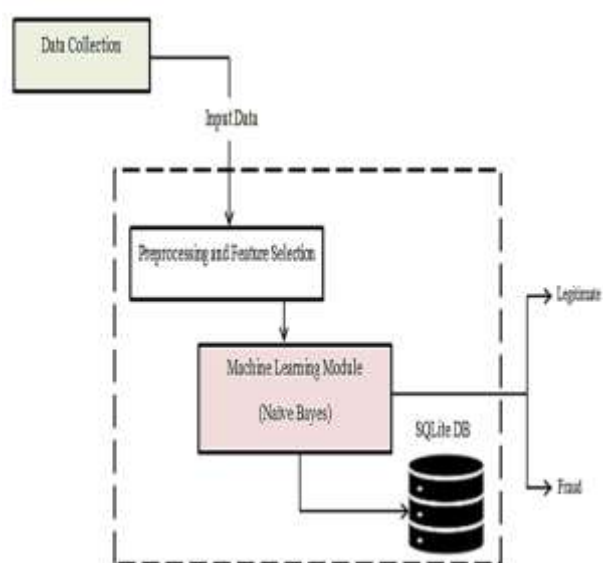


Figure 1: Architecture of the Proposed System

The architecture consists of five interconnected components. The Web Client is the browser-based front end through which users upload data, initiate analysis, and view

results. The Flask Application Server hosts the RESTful API that handles HTTP requests, orchestrates the preprocessing and classification pipeline, and manages authentication and session state. The Preprocessing Engine performs data cleaning, normalization, and feature engineering on incoming transaction data prior to inference. The Naïve Bayes Model is the trained classifier, serialized as a Python pickle file, loaded into memory at application start-up and used to score incoming transactions. The SQLite Database persists user accounts, transaction-batch records, classification results, and audit logs across sessions.

The Gaussian Naïve Bayes was selected as the specific variant for this study because PaySim’s transaction features are continuous and approximately normally distributed. The mathematical formulation used in the proposed system is presented below.

A. Bayes’ Theorem Foundation

Let $C \in \{0,1\}$ represent the transaction class, where $C = 0$ denotes a Legitimate transaction and $C = 1$ denotes a Fraudulent transaction. Let the feature vector describing a transaction be:

$$x = (x_1, x_2, \dots, x_n) \quad (1)$$

The Naïve Bayes classifier assigns the class $\hat{C}$ that maximizes the posterior probability:

$$\hat{C} = \arg \max_C P(C|x) = \frac{\arg \max_C [P(x|C) \times P(C)]}{P(x)} \quad (2)$$

Since $P(x)$ is constant across all classes, the decision rule is therefore simplified to:

$$\hat{C} = \arg \max_C P(C) \prod_i P(x_i|C) \quad (3)$$

This formulation assumes that all features are conditionally independent given the class label, which is the core assumption of the Naïve Bayes algorithm.

B. Prior Probability Estimation

The prior probability $P(C)$ represents the likelihood of each class occurring in the training dataset before any transaction features are observed, and computed as:

$$P(C = k) = \frac{N_k}{N} \quad (4)$$

Where N_k is the number of training samples belonging to class k , and N is the total number of training samples.

After applying the Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset, the estimated prior probabilities were approximately $P(\text{Fraud}) \approx 0.091$ and $P(\text{Legitimate}) \approx 0.909$.

These values indicate that legitimate transactions still constitute the majority class, although the imbalance was reduced after relative to the raw dataset.

C. Likelihood Estimation using Gaussian Distribution

The Gaussian Naïve Bayes classifier assumes that the continuous features follow a normal (Gaussian) distribution within each class.

For each feature x_i and class C , the likelihood probability is calculated using the Gaussian probability density function as follows:

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_{i,C}^2}} \times \exp\left(-\frac{(x_i - \mu_{i,C})^2}{2\pi\sigma_{i,C}^2}\right) \quad (5)$$

Where $\mu_{i,C}$ is the mean value of the feature x_i for class C and $\sigma_{i,C}^2$ is the variance of feature x_i for class C .

These statistical parameters are estimated from the training dataset during the fit() stage of the Gaussian Naïve Bayes model.

D. Log-Probability Computation

In practice, multiplying many small probabilities together may lead to numerical underflow. To prevent this issue, the classifier performs computations in logarithmic space. For this study, the posterior probability is therefore computed as:

$$\log P(C|x) \propto \log P(C) + \sum_i \log P(x_i|C) \quad (6)$$

The class with the highest log-posterior probability is selected as the predicted class.

To display the fraud probability in the system interface, the log-probabilities are converted back into normal probabilities using the softmax function:

$$P(C = 1|x) = \frac{\exp(\log P(C=1|x))}{\exp(\log P(C=0|x)) + \exp(\log P(C=1|x))} \quad (7)$$

This produces a normalized probability value between 0 and 1 representing the likelihood that a transaction is fraudulent.

E. Decision Threshold

The final classification decision is determined by comparing the predicted fraud probability against a predefined threshold θ . The classification rule in the proposed system is expressed as:

$$\text{Classification} = \begin{cases} \text{Fraud,} & \text{if } P(C = 1|x) \geq \theta \\ \text{Legitimate,} & \text{otherwise} \end{cases} \quad (8)$$

Finally, the implemented system’s threshold value is set to $\theta = 0.3$ instead of the conventional default value of 0.5.

This lower threshold was chosen to bias the classifier toward higher recall i.e. flagging more transactions as suspicious which is appropriate in a fraud-screening context where missed fraud is typically costlier than a false alarm. However, the specific value of 0.3 threshold is not derived from a formal cost-sensitivity analysis (e.g., a precision–recall trade-off curve or an explicit cost matrix) in this study, and future work should justify θ empirically rather than by convention.

F. Experimental Setup and Reproducibility

The PaySim dataset contains 6,362,620 simulated mobile-money transactions, of which 8,213 (about 0.129%) are labelled fraudulent. The data were divided into an 80% training set (approximately 5,090,096 transactions) and a 20% held-out test set (1,272,524 transactions) using a random split with random seed of 42. The test set therefore retains the natural class imbalance, with approximately 1,643 fraudulent transactions (about 0.13%) if the split was stratified.

The input features were the transaction type (encoded), amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest, stepHour. The selected features were normalized and then fitted on the training set only and applied unchanged to the test set.

SMOTE was applied after the train/test split and only to the normalized training data, so no synthetic record was generated from, or evaluated against, test transactions. For each synthetic sample, SMOTE selects a fraud record, chooses one of its k nearest fraud neighbours in feature space ($k = 5$), and creates a new point at a random position on the line segment joining the two. The sampling strategy was set to 0.1, meaning that synthetic fraud records were added until the fraud class reached 10% of the size of the legitimate class (approximately 508,353 fraud versus 5,083,526 legitimate training records). This yields the training priors $P(\text{Fraud}) \approx 0.091$ and $P(\text{Legitimate}) \approx 0.909$ reported in Section B; full 1:1 balancing was not used, which is why legitimate transactions remain the majority class. The SMOTE random seed was [SEED = 42], and the class means, variances and priors of the Gaussian Naïve Bayes model were estimated from this resampled training set with scikit-learn's default variance-smoothing value of 1×10^{-9} .

Nonetheless, for every test transaction the fraud posterior probability $P(C = 1 | x)$ in Equation (7) is computed, and the transaction is labelled fraudulent when this probability is at least $\theta = 0.3$ as shown in Equation (8). The same θ was applied to all test transactions and was fixed before the test-set evaluation. Because SMOTE changes the class priors and Gaussian Naïve Bayes posteriors are typically overconfident, the output is best interpreted as a fraud-risk score rather than a calibrated probability of fraud. The reported precision, recall and false positive rate apply only to $\theta = 0.3$; other thresholds move the classifier along the precision-recall trade-off, and a threshold-sensitivity analysis on a separate validation set is left for future work.

The fraud class is treated as the positive class with TP, TN, FP and FN denoting true positives, true negatives, false positives and false negatives respectively. Furthermore, $\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$; $\text{Precision} = TP / (TP + FP)$; $\text{Recall} = TP / (TP + FN)$; $\text{Specificity} = TN / (TN + FP)$; $\text{False Positive Rate} = FP / (FP + TN) = 1 - \text{Specificity}$; and $\text{F1-score} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall})$. AUC-ROC was computed from the continuous fraud posterior probabilities.

IV. RESULTS

The Naïve Bayes model was evaluated on a 20% held-out test set containing 1,272,524 transaction records. SMOTE oversampling was applied only to the training set to prevent data leakage. To validate the effectiveness of the developed SIM Fraud Detection System beyond batch evaluation, manual transaction testing was conducted using the real-time transaction classification interface. The interface allows users to manually input transaction parameters and obtain immediate fraud classification results.

Presented in Figure 2 and 3 is the screenshot interfaces of the manual transaction testing conducted for legitimate and fraudulent transactions respectively.

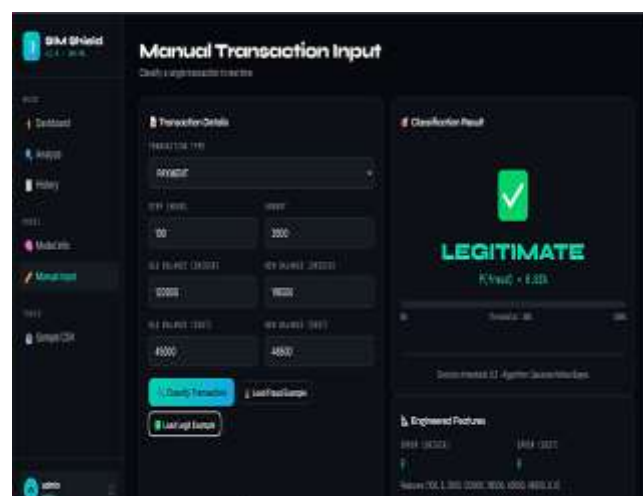


Figure 2: Transaction Classification Showing Legitimate Transaction Result

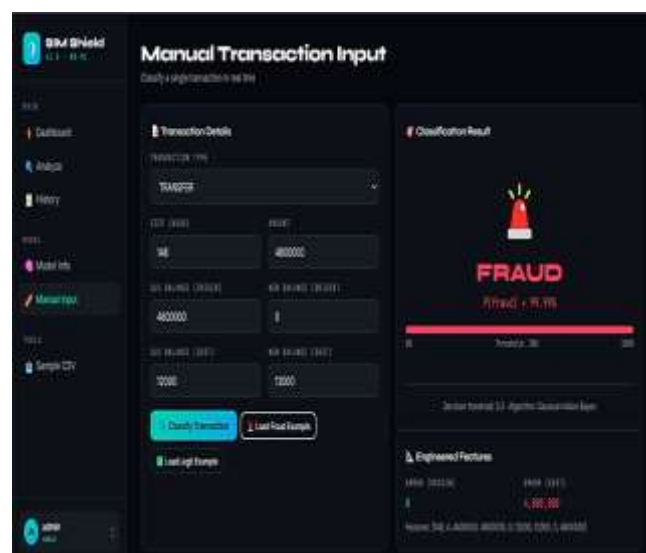


Figure 3: Transaction Classification Interface Showing Fraud Detection Result

Table 1 summarizes the performance metrics of the proposed Naïve Bayes-based fraud detection system on the held-out test set using PaySim dataset, and Table 2 gives the underlying confusion-matrix counts (TP, FN, FP and TN) from which every metric in Table 1 was computed using the

definitions in the Experimental Setup. Furthermore, Figure 4 captured a bar chart representation of the computed performance metrics.

Table 1 Naïve Bayes Performance Metrics on the held-out Test Set

Metric	Performance Value
Accuracy	99.91%
Precision	89.22%
Recall	88.67%
F1-Score	88.94%
Specificity	99.95%
AUC-ROC	0.9425
False Positive Rate	0.05%

Table 2 Confusion matrix counts on the held-out test set (fraud = positive class, $\theta = 0.3$)

Confusion Matrix	Count
True Positives (TP): fraud correctly flagged	4,736
False Negatives (FN): fraud missed	605
False Positives (FP): legitimate flagged as fraud	572
True Negatives (TN): legitimate correctly passed	1,266,611
Total test transactions (TP + FN + FP + TN)	1,272,524

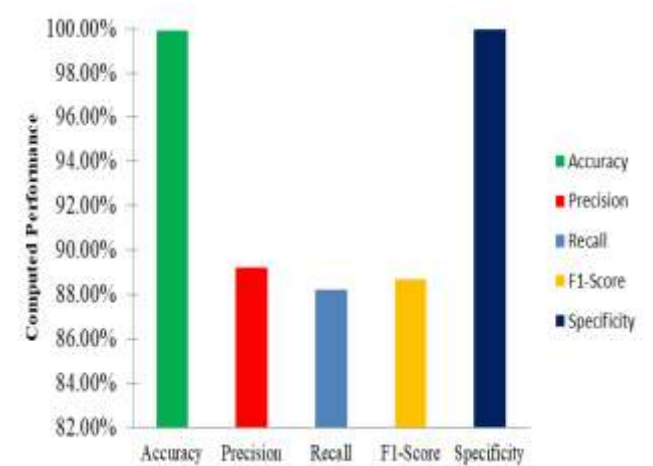


Figure 4: Bar Chart Representation of the computed Performance Metrics.

V. DISCUSSION

The experimental evaluation demonstrates the applicability of the proposed Gaussian Naïve Bayes classifier to transaction-level fraud screening using the PaySim synthetic mobile-money dataset. The model’s performance was assessed using accuracy, precision, recall, F1-score, specificity, false positive rate, and AUC-ROC. These metrics provide complementary perspectives on the classifier’s ability to distinguish fraudulent transactions from legitimate transactions.

The recall metric is particularly relevant to fraud detection because it represents the proportion of actual fraudulent transactions correctly identified by the classifier. Similarly, precision indicates the proportion of transactions classified as fraudulent that are genuinely fraudulent within the evaluation dataset. The consideration of both metrics is important because fraud detection systems must balance the identification of suspicious transactions against the operational consequences of false alarms.

The application of SMOTE exclusively to the training subset was intended to address class imbalance without contaminating the independent test evaluation. Retaining the original distribution of the test set provides a more appropriate basis for evaluating the classifier under the dataset’s natural class imbalance.

The use of Gaussian Naïve Bayes also provides computational advantages due to its relatively simple probabilistic estimation procedure. Nevertheless, the conditional-independence assumption may limit its ability to represent complex relationships among transaction attributes.

An important limitation concerns the nature of the PaySim dataset. Although it represents mobile-money transactions and includes fraudulent transaction labels, it does not contain direct telecommunications indicators such as SIM-swap events, SIM cloning, fraudulent SIM registration, device identity changes, or OTP interception patterns. Consequently, the results should be interpreted as evidence of mobile-money transaction fraud classification rather than direct detection of SIM-specific attacks.

Furthermore, because PaySim is synthetic, the reported performance cannot automatically be generalized to live mobile telecommunications environments. Real-world deployment would require validation using authentic mobile-money and telecommunications datasets containing relevant SIM-level behavioural and security attributes.

G. Comparison with Related Studies

Table 3 presents the performance metrics of the proposed model against the related works reviewed in this study.

Table 3 Indicative comparison of reported metrics across reviewed studies

Study	Method	Dataset	Reported Accuracy	Reported Recall
Proposed study	Gaussian Naïve Bayes	PaySim	99.91%	88.67%
Bah [11]	Random Forest	PaySim	99.95%	93.18%
Ibrahim <i>et al.</i> [12]	K-Means behavioural analytics	SA telecom operator	—	86.00%

“A Machine Learning Framework for Mobile Money Fraud Detection Using Gaussian Naïve Bayes: A Transaction-Level Evaluation on the Synthetic Paysim Dataset”

Husejinovic [13]	Naïve Bayes	Credit-card transactions	89.30%	—
Folorunso <i>et al.</i> [14]	LSTM	Nigeria telecom operator	94.70%	91.30%
Adewale <i>et al.</i> [15]	Random Forest (best)	PaySim	99.99%	—

VI. CONCLUSION

This study developed and evaluated a transaction-level mobile-money fraud detection framework using the Gaussian Naïve Bayes machine learning algorithm and the synthetic PaySim mobile transaction dataset. The study was motivated by the limitations of conventional rule-based fraud detection approaches, particularly their dependence on predefined rules and their difficulty in adapting to changing fraudulent transaction patterns. The developed system was implemented using a client-server architecture consisting of a web-based client interface, Flask application server, preprocessing engine, trained Gaussian Naïve Bayes model, and database component. The system also incorporated data preprocessing and SMOTE-based class balancing applied to the training set only (sampling ratio 0.1), and a decision threshold of 0.3 for converting fraud probabilities into class labels. The resulting framework provides both automated transaction classification and a manual interface through which transaction information can be submitted for immediate fraud assessment.

Experimental evaluation on the held-out PaySim test set demonstrated that the proposed model achieved 99.91% accuracy, 89.22% precision, 88.67% recall, and an F1-score of 88.94%. The model also achieved a specificity of 99.95%, a false-positive rate of 0.05%, and an AUC-ROC of 0.9425, all computed from the confusion-matrix counts in Table 2. Given the strong class imbalance of the dataset, recall and precision are more informative than accuracy. These results indicate that the proposed approach can distinguish fraudulent from legitimate PaySim transactions with a useful level of effectiveness at the chosen threshold of 0.3 while keeping the false-positive rate low.

The findings further demonstrate that the use of a relatively simple probabilistic machine-learning algorithm does not necessarily prevent the development of a useful fraud detection solution. Gaussian Naïve Bayes provides computational efficiency and probabilistic outputs that can be useful in transaction screening environments where large volumes of transactions require rapid assessment. The study therefore contributes to the development of an interpretable and computationally lightweight approach to mobile transaction fraud detection.

However, it is important to note that PaySim contains no SIM-specific attributes, so the study does not demonstrate the detection of SIM-related fraud as such. It is also important to note that the evaluation was conducted using the PaySim dataset, which is a synthetic representation of mobile money transactions. Therefore, the reported performance cannot automatically be assumed to represent the performance that would be obtained from a live telecommunications network containing real SIM-swap, SIM-cloning, fraudulent registration, and other SIM-related attack behaviours. In addition, the independence assumption underlying Naïve Bayes may not fully represent the relationships among transaction features. The reviewed literature also demonstrates that adaptive fraudsters can modify their behaviour in response to detection mechanisms, indicating that static evaluation alone may not adequately capture the challenges of real-world fraud detection.

Future research should therefore investigate the proposed framework using real-world telecommunications and mobile-money transaction data and evaluate its performance under changing and adversarial fraud behaviours. Future studies may also investigate deep learning and sequential models for capturing temporal transaction patterns, as well as real-time stream-processing mechanisms for continuous fraud monitoring. Hybrid approaches that combine the interpretability and computational efficiency of probabilistic models with more advanced machine-learning techniques may also be investigated. Such extensions could improve the adaptability of the proposed framework and provide a stronger basis for deployment in live telecommunications and mobile-money environments.

REFERENCES

1. P. Ashok and G. Hallur, "Transformational Trends Shaping the Future of the Telecom Industry," *Recent Advances in Electrical & Electronic Engineering*, vol. 18, pp. 1858-1878, 2025.
2. R. Deepthi, *New Era of Cyber Security: New Era of Cyber Security*: Geh press, 2026.
3. C. C. Cheruyot, R. M. Kouame, and H. Inaba, "Securing SIM Toolkit-based mobile money applications against SIM swap attacks using user location data," in *2024 IEEE 13th Global Conference on Consumer Electronics (GCCE)*, 2024, pp. 100-104.
4. R. Maity, "SIM Hijacking and Telecommunications Fraud: Attack Vectors, Impact, and Mitigation," *Cybersecurity Threats and Incident Response: Real-World Case Studies on Network Security, Data Breaches, and Risk Mitigation*, p. 28, 2025.
5. M. A. Hassan, Z. Shukur, and M. K. Hasan, "An improved time-based one time password authentication framework for electronic payments,"

- Int. J. Adv. Comput. Sci. Appl, vol. 11, pp. 359-366, 2020.
6. J. Scharfman, "The Cryptocurrency and Digital Asset Fraud Casebook, Volume II," Springer Books, 2024.
7. I. T. Sotonye and N. Eyidia, "Combating SIM Swap Fraud in Telecommunications: A Machine Learning Approach and Multi-Factor Authentication as a Preventive Strategy," Journal of Advancement in Communication System, vol. 8, 2025.
8. D. Njoku, V. Iwuchukwu, J. Jibiri, C. Ikwuazom, C. Ofoegbu, and F. Nwokoma, "Machine learning approach for fraud detection system in financial institution: A web base application," Machine Learning, vol. 20, pp. 01-12, 2024.
9. L. Ngartera, M. A. Issaka, and S. Nadarajah, "Hybrid naïve bayes models for scam detection: comparative insights from email and financial fraud," IEEE Access, 2025.
10. N. Baisholan, J. E. Dietz, S. Gnatyuk, M. Turdalyuly, E. T. Matson, and K. Baisholanova, "FraudX SimS: A Synthetic Dataset for Anomaly Detection in Payment-Card Transactions," IEEE Access, vol. 13, pp. 208549-208562, 2025.
11. M. Y. Bah, "Enhancing Mobile Money Security: A Multi-Layered Fraud Detection System Using Machine Learning and Multi-Factor Authentication," Journal of Software Engineering and Applications, vol. 19, pp. 131-153, 2026.
12. R. Ibrahim, O. O. Khalifa, and A. El-Hassan, "Behavioural analytics framework for SIM swap fraud detection," IEEE Access, vol. 9, pp. 56820-56833, 2021.
13. A. Husejinovic, "Credit card fraud detection using naïve Bayesian and c4. 5 decision tree classifiers," Husejinovic, A.(2020). Credit card fraud detection using naïve Bayesian and C, vol. 4, pp. 1-5, 2020.
14. O. Folorunso, A. Agboola, and V. E. Adeyemo, "Deep learning approaches to SIM fraud detection in telecommunications," Telecommunication Systems, vol. 82, pp. 213-228, 2023.
15. J. O. Adewale, O. O. Erastus, and V. A. Osekhonmen, "Development Of A Machine Learning-Based Anomaly Detection (MLAD) In Mobile Payment," Journal of Multidisciplinary Engineering Science Studies (JMESS), vol. 12, pp. 5913-5921, 2026.
16. A. El Qassouari, A. Charef, and Z. Jarir, "An Adaptive Multi-Agent Framework for Dynamic Fraud Simulation and Detection in Financial Transaction Networks," in 2026 6th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 2026, pp. 1-8.