International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No.42, January 2024 – www.ijais.org

Ensemble-based Predictive Model for Financial Fraud Detection

V.O. Olaleye
Department of Cybersecurity, School of Computing, Federal University of Technology Akure PMB 704, Akure, Ondo State Nigeria

O.A. Odeniyi, PhD
Department of Cybersecurity, School of Computing, Federal University of Technology Akure PMB 704, Akure, Ondo State Nigeria

B.K. Alese
Department of Cybersecurity, School of Computing, Federal University of Technology Akure PMB 704, Akure, Ondo State Nigeria

ABSTRACT
The financial industry remains a persistent target for fraudulent activities. Challenges to research in this area are due to data privacy concerns and the scarcity of publicly available datasets that contain instances of fraud. Researchers and practitioners have proposed various fraud detection techniques, applying diverse algorithms to uncover fraudulent patterns. To further address this, the study introduces a synthetic fraud-related dataset featuring five distinct fraud scenarios having about 2.5 million transactions. The primary objective is to analyze the intricacies of account transaction behaviour in a financial dataset. The authors propose an ensemble of three gradient boosting algorithms: CatBoost, Extreme Gradient Boosting (XGBoost), and LightGBM; The models developed demonstrate promising results, with several achieving an average Area Under the Curve (AUC) exceeding 0.9 and the ensemble having a predictive accuracy of 98.60%. Further evaluation through an application programming interface indicates a time complexity of less than 300 milliseconds and efficient memory usage, making this approach promising for practical usage in real-world scenarios.

General Terms
Data mining, Fraud Detection, Financial Industry

Keywords
Machine Learning, Synthetic Data, Financial Fraud, Ensemble Learning, Gradient Boosting.

1. INTRODUCTION
The financial industry has always been a target for fraudsters. A fraud detection system involving various detection techniques is essential for financial institutions to sustain goodwill from customers [1]. One barrier complicating the research in this direction is the lack of public data sets that embed fraudulent activities [2]. One of the ways banks can minimize their losses from fraud is through predictive modelling. Predictive modelling is a statistical analysis technique used to predict future events based on historical data. Using predictive modelling for fraud detection in the financial industry has been effective in improving fraud detection rates while reducing false positives.

A cohort of studies has effectively implemented these classifiers, exemplified by [3] who employed decision trees and artificial neural networks. [4] emphasized the pivotal role of financial data by applying a random forest classifier. Notably, ensemble learning methods have proven efficacious in this domain, as evidenced by the work of [5] Bian et al. (2018). In addition, Ensemble algorithms such as XGBoost, LightGBM and CatBoost have garnered recognition for their efficacy, with researchers like [6] Pesantez-Narvaez et al. (2019) attesting to their superiority over traditional decision trees, logistic regression when applied to telematics data. Financial institutions use Machine Learning to identify fraudulent patterns from large amounts of historical financial records. The detection of credit card fraud remains a significant challenge for business intelligence technologies as most datasets containing credit card transactions are highly imbalanced.

Another compelling aspect of predictive modelling is its capacity for continuous learning and improvement. [7] & [8] agreed that by leveraging historical data and machine learning algorithms, predictive models have the potential to enhance fraud detection accuracy, reduce false positives, and adapt to evolving fraud patterns.

Fraudsters are continually devising new methods to exploit vulnerabilities in financial systems. To effectively combat these evolving threats, fraud detection systems must be able to adapt and update their strategies in real-time [9]. Developing an effective financial fraud detection model presents a multifaceted challenge, particularly from a learning perspective. The intricacies lie in the dual hurdles of class imbalance, where genuine transactions significantly outnumber frauds (typically constituting less than 1% of transactions), and the presence of concept drift, wherein transactions change their statistical properties over time.

The prevalence of class imbalance, as documented in the literature [10, 11], adds a layer of complexity to the model development process. The rarity of fraudulent transactions makes it imperative to address the skewed distribution, ensuring the model remains adept at detecting fraudulent activities amidst the vast majority of legitimate transactions. Furthermore, the issue of concept drift, as highlighted in studies [12, 13], introduces a dynamic element to the problem. Transactions evolve over time, necessitating a model that can adapt to shifting statistical properties and remain resilient to emerging patterns of fraud.

A critical dimension of the challenge arises from verification latency, a phenomenon observed in real-world scenarios [14]. Verification latency entails a substantial delay in obtaining supervised samples, as professional investigators inspect alerts and engage with cardholders to ascertain the authenticity of each flagged transaction. This process yields valuable labelled transactions that can be used to train or

update the classifier. However, the majority of transactions elude verification due to time and cost constraints, leaving a gap in the feedback loop.

The literature review underscores a noteworthy gap in existing research, as many studies tend to overlook the crucial aspects of verification latency [15] and the alert–feedback interaction. This oversight neglects the real-world dynamics where investigators play a pivotal role in enhancing the model's performance over time by providing invaluable feedback through labelled transactions. The challenge lies in reconciling the need for timely verification with the practical constraints of resources, emphasizing the importance of addressing this aspect in the development of robust financial fraud detection models.

The main contributions of this paper are as follows:

1. The authors created a synthetic financial dataset with five fraud scenarios, addressing class imbalance using SMOTE and RandomUnderSampler to achieve a balanced dataset of 2,549,085 transactions.

2. This paper introduces an ensemble learning of three gradient boosting algorithms, leveraging its potential to outperform existing models in terms of detection accuracy and robustness.

3. With less than 400 milliseconds in time complexity, the ensemble model proves advantageous in reducing human verification latency, making it well-suited for scenarios where rapid and accurate fraud detection is imperative.

4. By having a space complexity of less than 3 megabytes, it ensures that the model is not resource-intensive, making it compatible with various computing environments without placing undue strain on system resources.

2. SCOPE AND METHODOLOGY

This research employs predictive modelling techniques using an ensemble machine learning approach, to enhance the detection of financial fraud. The ensemble method combines the predictions of three gradient-boosting algorithms to produce more accurate and robust results, making it particularly well-suited for fraud detection.

The research methodology as shown in the Figure 1 encompasses a cohesive series of stages. Initially, a comprehensive dataset was generated, inclusive of genuine and fraudulent financial transactions. The dataset before resampling initially encompassed approximately 1,800,000 transactions, which will collectively provide a substantial volume of data for comprehensive exploration. Subsequently, five fraud scenarios were introduced into the dataset to capture fraudulent instances.

Feature processing and engineering were systematically applied to extract pertinent information, enhancing the predictive capacity of the models. Following this, data preprocessing procedures, such as handling missing values, outlier detection, and data scaling, will be executed to prepare the dataset for model training. The training phase will involve the utilization of these gradient boosting algorithms, CatBoost, XGBoost, and LightGBM and leveraging their unique strengths to capture intricate patterns within the financial data. An ensemble approach was adopted, employing CatBoost as the meta-classifier, thereby amalgamating the predictive power of the individual models to enhance overall fraud detection accuracy. The research culminated in the development of an Application Programming Interface (API) for model deployment, seamless integration into financial systems, and a comprehensive evaluation involving a series of test cases to assess the ensemble model's efficacy and resilience in genuine financial fraud detection scenarios. The system methodology is shown below:

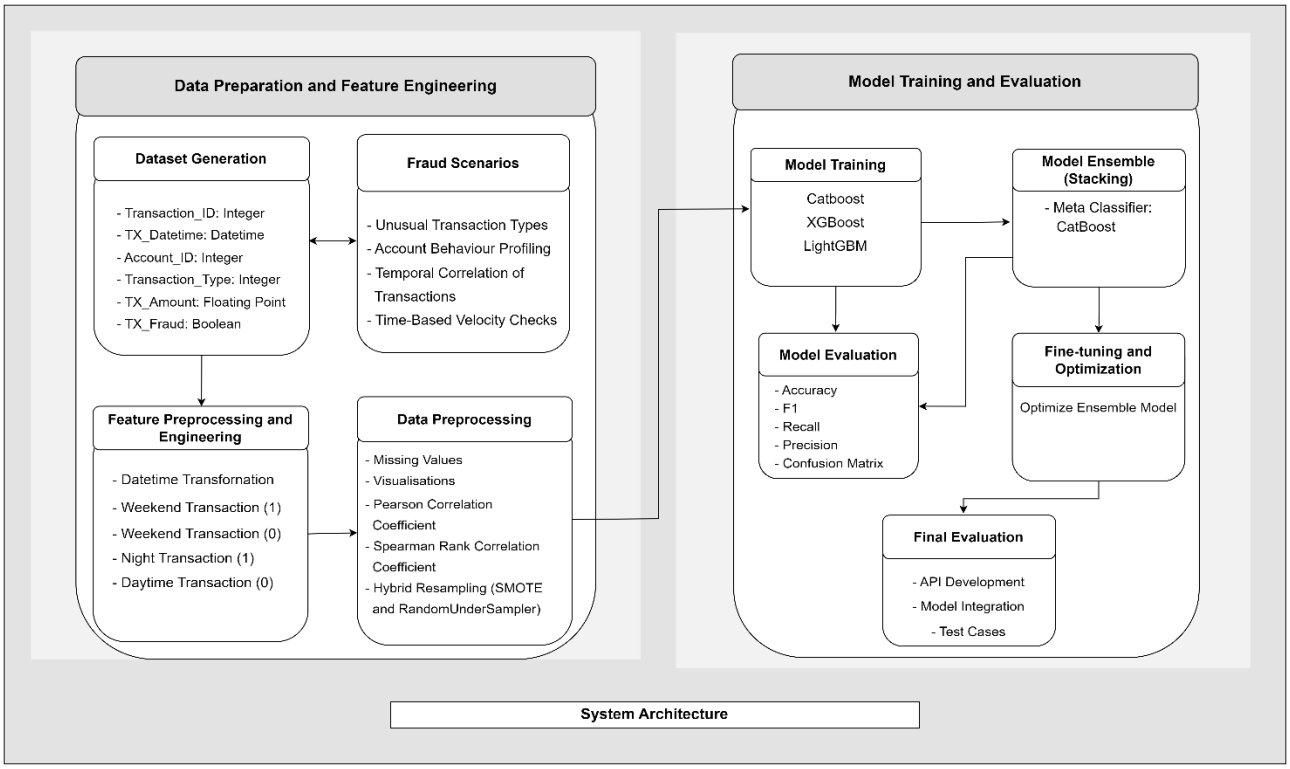

Figure 1: System Methodology

2.1 Dataset Generation

The process of dataset generation as shown in Figure 2 comprises three integral components: "Account Profiles," responsible for creating detailed account representations; "Transaction Generation," which simulates a diverse range of financial transactions; and the "Generation of Fraud Scenarios," which introduces predefined templates for identifying potentially fraudulent transactions based on various criteria. Together, these components collaboratively construct a dynamic and realistic dataset, mirroring real-world financial systems.

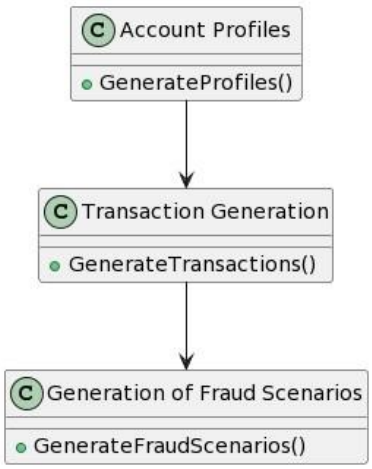

Fig. 2: Dataset Generation Process

2.1.1 Generation of Account Profiles

The account profile generation process uses randomization techniques to create a pandas DataFrame representing account profiles. Each account profile (shown in Figure 3) within this DataFrame is characterized by several key attributes, including a unique identification (ID), an arbitrary average transaction amount, the standard deviation of transaction amounts, the average daily transaction frequency and allowed transaction type. These attribute values are generated through a controlled randomization process.

	ACCOUNT_ID	mean_amount	std_amount	mean_nb_t
0	0	57137.282873	28568.641437	
1	1	62262.520727	31131.260363	
2	2	45247.205937	22623.602969	
3	3	46570.785070	23285.392535	
4	4	96547.962248	48273.981124	

Fig. 3: Account Profile Sample Data

2.1.2 Transaction Generation

The transaction generation process systematically simulates the creation of a comprehensive transaction dataset, it serves to produce a structured table of transactions associated with existing customer account profiles. This generation task is geared towards creating transaction records for a substantial dataset encompassing 5000 unique account IDs, spanning a period of 183 days.

	TX_DATETIME	ACCOUNT_ID	TRANSACTION_TYPE	TX_AMOUNT
0	2023-04-01 07:19:05	0	2	113415.35
1	2023-04-01 19:02:02	0	2	42685.08
2	2023-04-01 18:00:16	0	1	70973.42
3	2023-04-02 15:13:02	0	2	29689.83
4	2023-04-02 14:05:38	0	2	58092.59

Fig. 4a: Sample Transaction Dataset

Following the sample transaction data in Figure 4a, each individual transaction record generated within this dataset is characterized by four essential attributes: timestamp, account ID, transaction type, and amount. The timestamp provides temporal context, indicating when each transaction occurred, while the account ID has a linkage between transactions and their respective accounts. The transaction type specifies the nature of each financial transaction, whether it's an ATM Withdrawal (1), Teller Withdrawal (2), or Online transaction (3). Meanwhile, the amount attribute records the monetary value involved in each transaction, signifying the currency associated with it.

To achieve the transaction generation, the process considers several factors which firstly determine the daily transaction frequency probabilistically by drawing from a Poisson distribution. The mean value for this distribution is derived from the corresponding account profile, ensuring a realistic representation of transaction frequencies.

Also, the timing of each transaction is chosen randomly, with a preference for transactions to occur around noon. This temporal distribution is created by drawing from a normal distribution centred at that specific time, mirroring natural timing patterns commonly observed in real financial transactions. Finally, the amount of each transaction is generated through a random process. Specifically, it is drawn from a normal distribution using parameters obtained from the account profile, including the mean and standard deviation. In cases where a negative amount is generated, the process ensures realism by redrawing the amount from a uniform distribution, ensuring that all transaction amounts are positive. A sample of the resulting output of these processes is shown in the figure below.

	TX_DATETIME	ACCOUNT_ID	TRANSACTION_TYPE	TX_AMOUNT	TX_TIME_SECONDS	TX_TIME_DAYS
0	2023-04-01 07:19:05	0	2	113415.35	26345	0
1	2023-04-01 19:02:02	0	2	42685.08	68522	0
2	2023-04-01 18:00:16	0	1	70973.42	64816	0
3	2023-04-02 15:13:02	0	2	29689.83	141182	1
4	2023-04-02 14:05:38	0	2	58092.59	137138	1

Fig. 4b: Transaction Dataset

2.1.3 Data Distribution

Figures 5a and 5b depict the distribution of transaction amounts and times respectively. They provide insight into the spread and frequency of transaction values and aid in the identification of outliers. The distribution of transaction times indicates that a larger volume of transactions occurs during mid-day which is useful when simulating fraudulent instances.

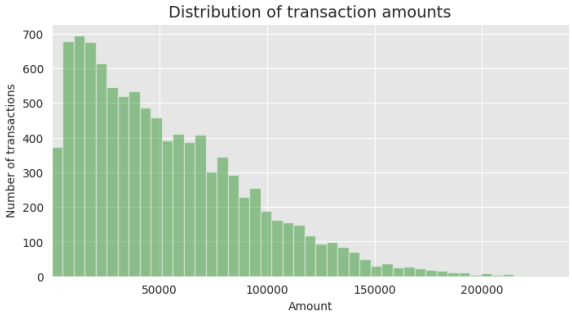

Fig. 5a: Distribution of transaction amount

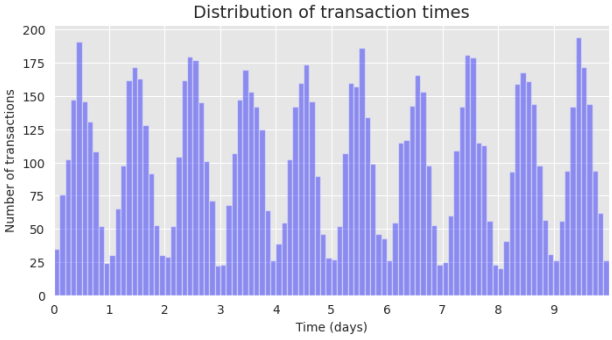

Fig. 5b: Distribution of transaction times

2.1.4 Fraud Scenarios Generation

Each fraud scenario devised in this research is designed to capture specific types of fraudulent behaviour. These scenarios leverage transaction type, amount, timing, and historical account behaviour to detect anomalies that may indicate fraudulent transactions. These fraud scenarios collectively form a comprehensive framework for identifying potential fraudulent transactions within the generated dataset.

Scenario 1: Unusual Transaction Types

This scenario focuses on identifying potentially fraudulent transactions based on unusual transaction types. Specifically, it flags transactions with transaction types that differ from the norm, specifically those that are not categorized as 1, 2, 3 or not trained with the account data. By flagging transactions with non-standard types, it aims to capture potentially suspicious activities that deviate from typical transaction behaviour on the account.

Scenario 2: Account Behaviour Profiling

In this scenario, potential fraudulent transactions are identified by comparing the transaction amount to the account's historical behaviour. Transactions with amounts that significantly deviate from the mean amount for the account (greater than twice the standard deviation) are flagged. By examining the historical behaviour of an account, this scenario seeks to detect unusual transaction amounts that may indicate fraudulent activity. Significant deviations from the account's typical spending patterns are marked as fraudulent.

Scenario 3: Temporal Correlation of Transactions

This scenario is concerned with the temporal aspects of transactions. It flags transactions where the time difference between consecutive transactions made by the same account is less than 5 seconds as potential fraud. Rapid consecutive transactions within an exceedingly short time frame suggest unusual and potentially fraudulent behaviour. Such a quick succession of transactions are marked as fraudulent.

Scenario 4: Time-Based Velocity Checks

Similar to Scenario 3, this scenario focuses on the timing of transactions. It identifies potential fraud by flagging transactions where the time difference between consecutive transactions for the same account is less than 60 seconds. A feature calculates the time difference between consecutive transactions for each account, if a sudden spike is observed in transaction velocity, it is marked as fraudulent activity. Rapid transactions within a short time frame can also indicate suspicious activity, as it may be an attempt to quickly make multiple transactions before being detected. This scenario aims to capture such velocity-based anomalies.

Scenario 5: 1-Hour Window Analysis

This scenario considers both the timing and the amount of transactions within a specific time window. It identifies potential fraud by flagging transactions occurring within a 1-hour window where the z-score (standardized deviation from the mean) of the transaction amount exceeds a predefined threshold. This scenario combines temporal and amount-based analysis to detect unusual spikes in transaction amounts within a short time frame. High z-scores indicate transactions that significantly deviate from the norm, potentially indicating fraudulent activity within that specific time window.

2.1.5 Dataset Preprocessing

To ensure that the model is exposed to a more balanced dataset, reducing the risk of biased predictions and enhancing the model's ability to accurately classify both legitimate and fraudulent transactions. A hybrid data resampling technique was employed, combining the Synthetic Minority Over-sampling Technique (SMOTE) and RandomUnderSampler. The objective was to address the class imbalance in the dataset, where the number of legitimate transactions (Class 0) far exceeded that of fraudulent transactions (Class 1).

The initial class distribution revealed a significant class imbalance, with 1,699,391 instances of legitimate transactions (Class 0) and only 55,893 instances of fraudulent transactions (Class 1). The hybrid resampling technique aimed to rectify this imbalance. After applying SMOTE and RandomUnderSampler, the class distribution was modified to 1,699,390 instances of legitimate transactions (Class 0) and 849,695 instances of fraudulent transactions (Class 1). The entire dataset consists of 2,549,085 transactions.

2.2 Model Training

The training of the fraud detector for each base learner was carried out on 10 epochs.

Base Learner 1: Categorical Boosting Algorithm (CatBoost)

CatBoost is known for its robust gradient-boosting capabilities and is selected as one of the individual base learners. Its intrinsic support for categorical features, adept handling of missing data, and efficient training process make it a valuable asset for predictive modelling to be used in this research. CatBoost's prediction for a given instance x can be expressed as:

$$\hat{y}_i = offset + \sum_{t=1}^{T}(scale \cdot tree_t(x_i)) \qquad (1)$$

Where:

- $\hat{y}_i$ is the predicted output for instance x_i
- T is the total number of trees in the ensemble.
- $tree_t(x_i)$ represents the output of the t-th tree for instance.
- offset and scale are constant values used for adjusting the final prediction.

Base Learner 2: Light Gradient Boosting Machine (LightGBM)

LightGBM is a high-performance gradient boosting framework and is chosen as another individual base learner for its distributed and efficient training, as well as its ability to handle large datasets, LightGBM provides for complex predictive tasks and has a potent gradient-boosting technique and operates by iteratively refining predictive models through the aggregation of numerous weak learners, commonly represented as decision trees. LightGBM is notable for its efficiency in handling large datasets and its capability to optimize memory usage.

Base Learner 3: Extreme Gradient Boosting Algorithm (XGBoost)

XGBoost is an optimized and scalable gradient boosting library and is integrated as the third individual base learner. Leveraging regularization techniques and parallel processing capabilities, XGBoost contributes to accurate and efficient model training. XGBoost is mathematically expressed as:

F_x= *Loss function + Regularization* (2)

$F_x = [\sum_{i=0}^{n} L(yi, pi + Ovalue)] + \frac{1}{2}\lambda O2value$ (3)

$Ovalue = \frac{Sum\ of\ Residuals}{Numbers\ of\ Residuals+\lambda} = \frac{(g1+g2+\cdots gn)}{(h1+h2+\cdots hn+\lambda)}$ (4)

$L\ (yi, pi) = ½\ (y_i - p_i)^2$ (5)

Where*:*

L= loss function

H = hessian (hessian is a square matrix of second-order partial derivatives of a scalar value function)

g = gradient

λ = lambda

y_i *= observed value*

p_i*= previous probability*

Ovalue = output value

The Ensemble Model

The three individual base learners are combined into a stacked ensemble. The stacked model capitalizes on the diverse strengths of each individual model, fostering a synergistic relationship that has the potential to enhance predictive performance. The ensemble approach is designed to exploit the complementary nature of three individual base learners, CatBoost, LightGBM, and XGBoost, aiming for improved predictive accuracy and robustness. The stacked ensemble configuration, with CatBoost as the meta-classifier, is anticipated to exhibit a better understanding of the data and potentially outperform the individual base learners. Each individual base learner is trained on the training set, optimizing its parameters for effective learning. The ensemble model is subsequently trained on the same training set, leveraging the predictions of the individual models to enhance overall performance.

3. RESULTS AND DISCUSSION

The performance analysis of the models was evaluated using robustness analysis and complexity analysis.

3.1 Robustness Analysis

The robustness analysis measures the model's inherent robustness within the financial dataset and it is evaluated using the following metrics:

The *accuracy* metric given in Equation (6) represents the proportion of all predictions that are correct.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

The *precision* metric given in equation (7) is the proportion of predicted positives that are actually positive.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

The *recall* metric is given in Equation (8)

$$\text{Recall} = \frac{TP}{TP+FN} \quad (8)$$

The *f1-score* metric is given in Equation (9)

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (9)$$

The robustness analysis table is presented in table 1

Table 1: Models Robustness Analysis

Model	Accuracy	Recall	Precision	F1
CatBoost	97.92%	97.15%	96.63%	96.89%
XGBoost	98.51%	97.09%	98.42%	97.75%
LightGBM	98.39%	97.41%	97.76%	97.59%
Ensemble	98.60%	97.15%	98.62%	97.88%

Considering the table above, XGBoost demonstrates superiority in terms of precision, recall, and F1 score among the base learners. This suggests that XGBoost has a good balance between making accurate positive predictions (precision), capturing a high proportion of actual positive instances (recall), and harmonizing these aspects in the F1 score. CatBoost, while exhibiting slightly lower precision and recall compared to XGBoost, compensates with high accuracy and F1 score. Although it may not achieve the same level of precision and recall, it maintains a strong overall performance, especially in terms of accurately classifying instances and

achieving a balance between precision and recall. LightGBM consistently performs competitively across metrics. While it may not surpass XGBoost or the Ensemble in specific aspects, it maintains a balanced and robust performance. This suggests that LightGBM is a reliable model across various evaluation criteria. The confusion matrixes (Fig. 7 – 9) provide insights into the performance of a classification model. It shows the true positive, true negative, false positive, and false negative values, allowing us to assess the model's accuracy, precision, recall, and F1 score. These insights help in understanding the model's strengths and weaknesses.

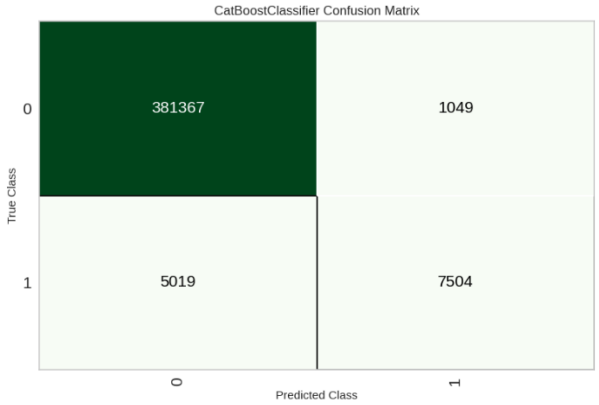

Fig. 6 Confusion Matrix of the Base-learner 1: CatBoost

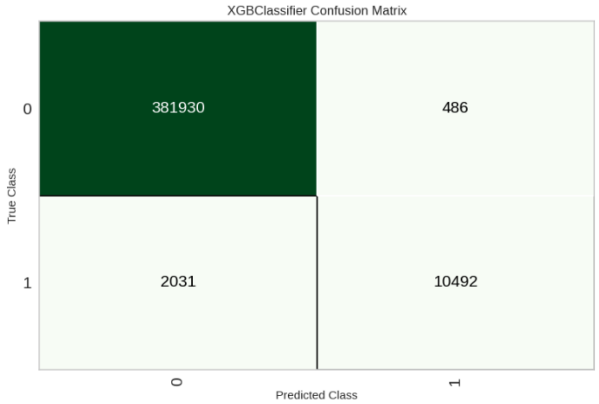

Fig. 7 Confusion Matrix of the Base-learner 2: XGBoost

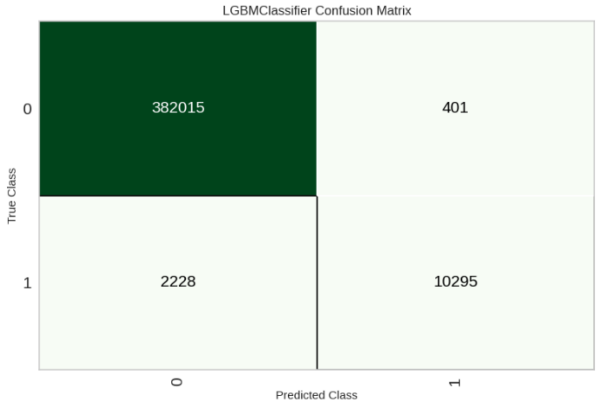

Fig. 8 Confusion Matrix of the Base-learner 3: LightGBM

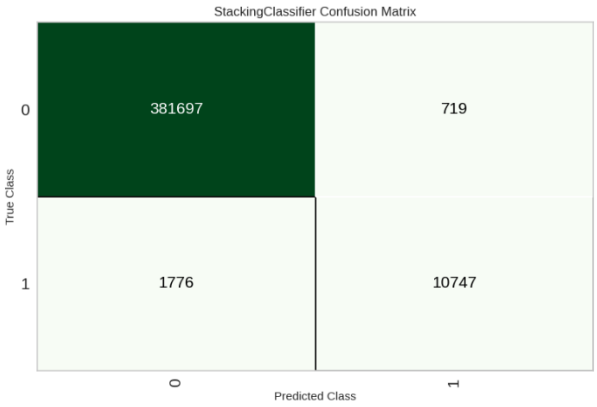

Fig. 9 Confusion Matrix of the Ensemble

To compare the confusion matrices of the models, we can derive the following insights:

True Positives (TP):

The StackingEnsemble and XGBClassifier models have the highest number of true positives, indicating their ability to correctly identify fraudulent transactions.

CatBoost and LGBClassifier also perform well in this aspect but have slightly fewer true positives compared to XGBClassifier and StackingEnsemble.

True Negatives (TN):

All the models have high numbers of true negatives, indicating their ability to accurately identify non-fraudulent transactions.

False Positives (FP) and False Negatives (FN):

The LGBClassifier model has the lowest number of false positives, indicating a better ability to avoid misclassifying non-fraudulent transactions as fraudulent.

The XGBClassifier model also performs well in this aspect.

The StackingEnsemble has the lowest number of false negatives, indicating its better ability to correctly classify fraudulent transactions.

Concluding on the confusion matrices, the StackingEnsemble Classifier appears to outperform the individual classifiers in detecting fraudulent instances, as it has the highest number of true positives and true negatives, and the lowest number of false positives and false negatives. By complementing its accuracy, it also shows balanced performance across precision, recall, and F1 score. The Ensemble model is designed to mitigate the weaknesses of individual learners, resulting in a more robust and effective predictive model.

3.2 Complexity Analysis

This research further implements an Application Programming Interface (API) to assess the time complexity computational cost associated with the predictive model and frameworks utilized. The implementation of the API provides an avenue for a comprehensive assessment of the computational costs associated with deploying the final model. This practical evaluation is crucial for determining the feasibility and efficiency of integrating the fraud detection model into real-world applications. It also allows for a more informed consideration of the trade-offs between accuracy and

computational efficiency in deploying such models in diverse and dynamic environments.

3.2.1 API Computational Evaluation

Table 2: Complexity Analysis

Time Complexity (Milliseconds)	Space Complexity
< 400	3mb

The implementation of an API serves as a crucial step towards understanding the real-time computational cost associated with deploying the final model. Generally, it achieved a time complexity of less than 400 milliseconds and a space complexity of less than 3 megabytes during testing. The efficient time complexity ensures that the API, when deployed in a live environment, can respond swiftly to requests, making it suitable for applications that demand quick and responsive fraud detection. Also, having a space complexity of less than 3 megabytes, it ensures that the API is not resource-intensive, making it compatible with various computing environments without placing undue strain on system resources. This section details some testing scenarios for the API. the testing scenarios will be divided into two three categories:

i. Unusual Transaction Type
ii. Account Behaviour Profiling
iii. Temporal Correlation of Transactions

3.2.1.1 API Test Case 1 - Unusual Transaction Type

The primary objective of this API testing case is to verify that the API, connected to the ensemble model, correctly identifies and flags a transaction as fraudulent when it falls outside the allowed or usual transaction types on the account. This scenario involves submitting a transaction with specific characteristics to the API, connected to the ensemble model, and ensuring the system correctly detects it as fraudulent due to an unusual transaction type.

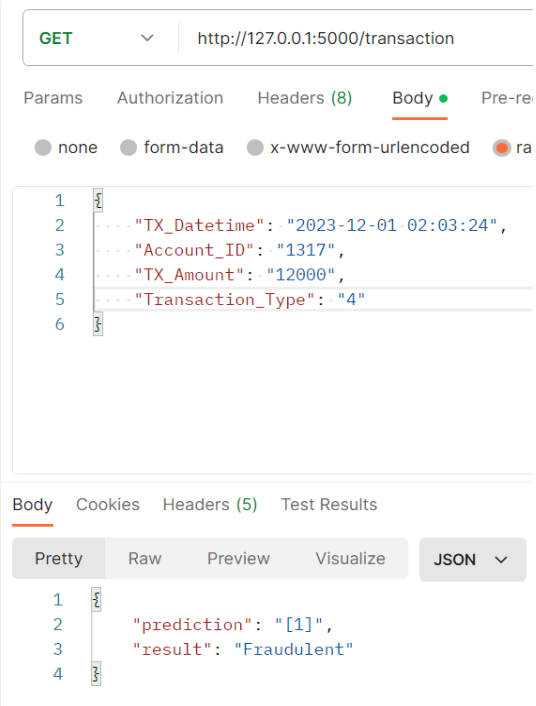

Fig. 10 Case 1 - Unusual Transaction Type

A request is sent to the API with the provided test data, the anticipated outcome was that the API, integrated with the ensemble model, would successfully detect and flag the test case as fraudulent. Upon the execution of the test case which took about 336 milliseconds, the API successfully flagged the transaction as fraudulent. The API response included a 'Fraudulent' status, and further analysis revealed that the ensemble model, connected to the API, accurately identified the unusual transaction type and made the appropriate decision.

3.2.1.2 API Test Case 2 - Account Behaviour Profiling

The objective of this test case was to assess the API's ability to profile account behaviour and identify anomalies, specifically focusing on a scenario where the transaction amount significantly deviates from the account's historical behaviour. The scenario involved submitting a transaction with a substantial amount to the API, simulating a case where the transaction amount completely deviates from the account's historical behavior. The goal was to verify if the API correctly identifies and flags such anomalous transactions as fraudulent.

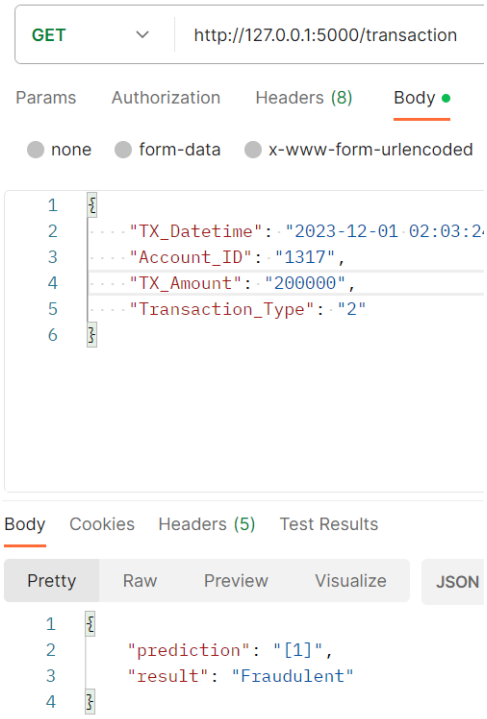

Fig. 10 Case 2 - Account Behaviour Profiling

As shown in Figure 10, a request was sent to the API with the specified test data. The anticipated outcome was that the API, leveraging account behaviour profiling, would successfully detect and flag the test case as fraudulent due to the significant deviation in the transaction amount from the account's historical behaviour. The API successfully flagged the transaction with a #200,000 amount as fraudulent. The explanation behind this conclusion is that Account ID 1317 has a mean and standard deviation of #70,990 and #32,384 respectively as contained in the dataset. The transaction deviates from the account threshold of 64,768 by 135,232. The API response included a 'Fraudulent' status, and further analysis confirmed that the account behaviour profiling mechanism correctly identified the anomaly, taking into account the historical mean, standard deviation, and the established amount threshold.

3.2.1.3 Case 3: Temporal Correlation of Transactions

The objective of this test case was to evaluate the API's capability to detect temporal correlations among consecutive transactions, particularly in a scenario where the transaction amount is consistent but deviates from the established threshold. The test involved running transactions consecutively for nth times, with the fraudulent flag triggered at the 5th transaction due to temporal correlations. This scenario in figure 11 simulated a sequence of consecutive transactions for the same account, each with an amount of #12,000. The goal was to assess whether the API, when considering temporal correlations, could correctly flag the nth transaction that has exceeded the threshold within a timeframe as fraudulent, given the established account mean, standard deviation, and amount threshold.

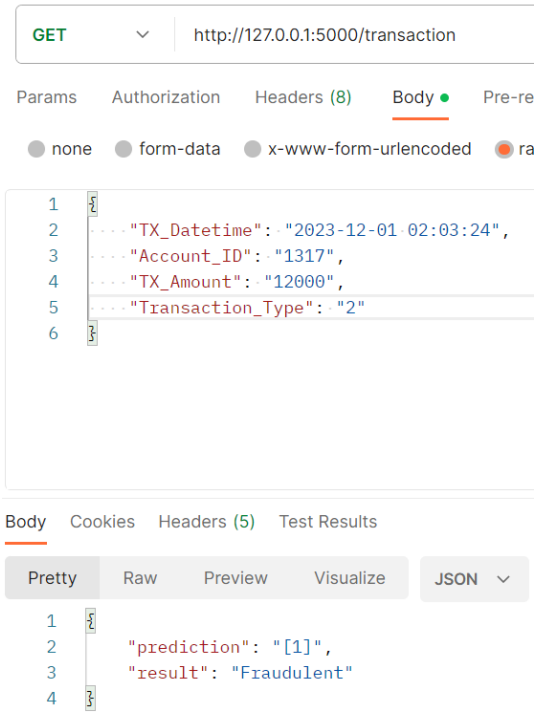

Fig. 11 Case 3: Temporal Correlation of Transactions

A sequence of consecutive transactions with an amount of #12,000 each was submitted to the API for the same account. The anticipated outcome was that the API, considering temporal correlations, would correctly flag the 5th transaction as fraudulent while treating the preceding transactions as non-fraudulent. The API successfully identified temporal correlations among the consecutive transactions and flagged the 5th transaction as fraudulent. The API responses aligned with the expected behaviour, with the temporal correlation detection mechanism triggering the fraudulent flag at the appropriate point in the sequence.

4. CONCLUSION

This research further addressed the challenge of fraud detection by defining five fraudulent instances and developing a detection model capable of identifying patterns related to fraud within a dataset. The scarceness of evidence for real-world fraudulent instances necessitated the creation of a synthetic dataset, which underwent preprocessing and resampling using a Hybrid strategy (SMOTE and RandomUnderSampler). The resulting dataset consists of 2,549,085 records (1,699,390 non-fraud instances and 849,695 fraud instances). The distinctiveness of this study is evident in its integration of three ensembled boosting algorithms with the substantial dataset containing an ample number of fraudulent instances. This approach effectively identifies fraud-related behaviour in accounts across five predefined fraudulent scenarios with 98.60% predictive accuracy; 97.15% Recall; 98.62% in Precision; 97.88% in F1 Score and an AUC of 99%.

To the best of the authors' knowledge, this methodology has not been employed as a reference framework in any existing research. Consequently, this innovative approach may serve as a foundation for tackling this issue from diverse perspectives, particularly by integrating it in financial institutions. The demonstrated efficacy of this approach underscores its promise as a viable solution for mitigating fraudulent transactions in the financial sector. With predictive accuracies exceeding 98.60% and efficient processing characteristics such as a time complexity of less than 300 milliseconds and optimized memory usage, this approach showcases practical utility for real-world applications. Financial institutions can confidently consider adopting this methodology, leveraging the ensemble of gradient-boosting algorithms, to bolster their fraud detection capabilities.

This contribution opens avenues for further exploration and development in the field of financial fraud detection.

5. ACKNOWLEDGEMENTS

The authors acknowledge the Rockshield Microfinance Bank LTD and the Cybersecurity department of the Federal University of Technology, Akure.

6. DATA AVAILABILITY STATEMENT

The dataset generated to support the findings of this study is accessible on request. interested researchers can request access to the code by sending an email to the corresponding author via (ictoluseyi@gmail.com).

7. REFERENCES

[1] D. Prusti and S. K. Rath, "Fraudulent Transaction Detection in Credit Card by Applying Ensemble Machine Learning techniques," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944867.

[2] Sánchez-Aguayo, M., Urquiza-Aguiar, L., & Estrada-Jiménez, J. (2022). Predictive Fraud Analysis Applying the Fraud Triangle Theory through Data Mining Techniques. Applied Sciences, 12, 3382. https://doi.org/10.3390/app12073382

[3] Paefgen, J., Staake, T., & Thiesse, F. (2013). Evaluation and aggregation of pay-as-you-drive insurance rate factors: A classification analysis approach. Decision Support Systems, 56, 192–201

[4] Baecke, P., & Bocca, L. (2017). The value of vehicle telematics data in insurance risk selection processes. Decision Support Systems, 98, 69–79.

[5] Bian, Y., Yang, C., Zhao, J. L., & Liang, L. (2018). Good drivers pay less: A study of usage-based vehicle insurance models. Transportation Research Part A:

Policy and Practice, 107, 20–34.

[6] Pesantez-Narvaez, J., Guillen, M., & Alcaniz, M. (2019). Predicting motor insurance claims using telematics data—xgboost versus logistic regression. Risks, 7(2), 70.

[7] Prates, J. M., Oliveira, L. S., Costa, K. A., & Ludermir, T. B. (2011). Predictive modelling for fraud detection: A data-oriented approach. Decision Support Systems, 51(1), 201-210.

[8] Geetha, G., Navin, J., Sanjeevi, P., & Sivaraj, S. (2023). Driver Driving Performance Analysis And Risk Detection Using Deep Learning. International Journal of Advanced Research in Computer and Communication Engineering, 12(5), 388–394. https://doi.org/10.17148/IJARCCE.2023.12563

[9] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi,"Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy" in IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, 2015, pp. 1–14.

[10] A. Dal Pozzolo, O. Caelen, and G. Bontempi, "When is undersampling effective in unbalanced classification tasks?" in Machine Learning and Knowledge Discovery in Databases. Cambridge, U.K.: Springer, 2015

[11] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in Proc. IEEE Symp. Ser. Computat. Intell., Dec. 2015, pp. 159–166

[12] C. Alippi, G. Boracchi, and M. Roveri, "Just-in-time classifiers for recurrent concepts," IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 4, pp. 620–634, Apr. 2013.

[13] J. Gama, I. Žliobait˙ e, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Comput. Surv., vol. 46, no. 4, p. 44, 2014.

[14] G. Krempl and V. Hofer, "Classification in presence of drift and latency," in Proc. 11th Data Mining Workshops, Dec. 2011, pp. 596–603.

[15] J. Plasse and N. Adams, "Handling delayed labels in temporally evolving data streams," in Proc. Int. Conf. Big Data, 2016, pp. 2416–2424.