www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Fraud Shield: Detecting Online Payment Fraud Using Machine Learning

[1]Dr.C.Srinivas Kumar
Professor and Dean, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
email:drcskumar46@gmail.com

[2]M.Sriharsha
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
email:sriharshamanneru@gmail.com

[3]B.Ravali
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
email:gravali919@gmail.com

[4]P.Usha
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
email:ushasarvani94@mail.com

Abstract— Online payment fraud is a growing concern in digital financial transactions. This study presents a machine learning- based solution to detect fraudulent transactions using supervised classification techniques. The project utilizes a publicly available dataset containing transaction features and labels (fraudulent or legitimate). The goal is to build a predictive model capable of distinguishing between fraudulent and non-fraudulent activities with high accuracy and minimal false positives. The rapid growth of online payment platforms has led to an ever-increasing volume of transactions, proportionately raising the risk of fraudulent activities. This paper presents a comprehensive study on detecting payment fraud using supervised machine learning techniques implemented in Python. We explore data preprocessing strategies, feature engineering methods, and the comparative performance of several classification algorithms, including Logistic Regression, Random Forest, Gradient Boosting, and XGBoost. The dataset comprises anonymized transaction records containing numerical and categorical attributes, which are preprocessed through normalization, one-hot encoding, and class imbalance handling using SMOTE. Model training and evaluation are performed using stratified k-fold cross-validation, and performance metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) are reported. The best-performing model, XGBoost, achieves an AUC-ROC of 0.982 and a recall of 0.935, indicating its suitability for real-time fraud detection systems. Finally, we discuss deployment considerations, such as inference latency and integration with production payment gateways, and outline future work on deep learning and online learning approaches.

keywords—Fraud Detection, Machine Learning, Logistic Regression, Python, Online Payments, Supervised Learning

I. INTRODUCTION

Online payment systems have revolutionized commerce by enabling fast and convenient transactions. With the increasing adoption of digital wallets, mobile banking apps, and e-commerce platforms, the volume of online transactions has surged dramatically.

As these platforms expand, they become increasingly attractive to cyber criminals who exploit system vulnerabilities to commit financial fraud, identity theft, and unauthorized account access. Payment fraud not only results in substantial monetary losses but also erodes consumer trust and imposes legal and reputational risks on financial institutions. Traditional rule-based fraud detection systems, although still in use, often struggle to adapt to rapidly evolving fraud tactics. These systems rely on predefined patterns and thresholds, making them prone to high false positives and ineffective against sophisticated, subtle fraud schemes. In contrast, data- driven approaches using machine learning offer dynamic, adaptive solutions capable of identifying complex, non-linear relationships within large volumes of transaction data. Detecting fraudulent behavior in payment data is challenging due to the highly imbalanced nature of transaction classes, evolving fraud patterns, and the need for low-latency decision- making. Moreover, fraudsters continuously evolve their strategies to evade detection, necessitating the development of intelligent systems that can learn from historical data and adapt over time. Machine learning (ML) offers powerful tools to model complex relationships in transactional data and flag potentially fraudulent activities before they impact merchants or consumers. to model complex relationships in transactional data and flag potentially fraudulent activities before they impact merchants or consumers.
In this work, we develop and evaluate ML-based classifiers using Python for online payment fraud detection. Our contributions include: (i) a detailed preprocessing pipeline tailored to transaction data; (ii) a comparative analysis of multiple ML algorithms; (iii) deployment guidelines for integration into streaming payment workflows The exponential rise in digital payment systems has opened new avenues for financial transactions but also exposed users to significant risks, particularly fraud. Online payment frauds result in substantial financial losses and undermine user trust in digital financial platforms. Traditional rule - based systems often lack the flexibility to detect evolving fraud tactics.

In this context, machine learning (ML) offers a powerful alternative by learning patterns in historical data and dynamically adapting to new fraudulent behaviors. This study presents an ML-based approach using Logistic Regression to detect fraudulent transactions in online payments, leveraging transaction features for accurate classification.

II. LITERATURE REVIEW

Prior studies have applied statistical and ML techniques to fraud detection. Bolton and Hand introduced unsupervised methods for credit card fraud using peer group and clustering approaches. Whitrow et al.demonstrated the effectiveness of supervised classifiers with feature aggregations over time windows. More recent works employ ensemble methods such as Random Forest and Gradient Boosting,achieving high detection rates but often at the cost of increased inference time. Deep learning models, including auto encoders and recurrent neural networks, have shown promise but require extensive tuning and computational resources.

Fraud detection has been extensively researched in recent years. Sahin and Duman[1]used decision trees and support vector machines to detect credit card fraud, showing the efficacy of ML in this domain.Dal Pozzolo et al.[2] addressed imbalanced datasets using ensemble learning to enhance fraud detection. Roy. et al.,[3] conducted a comparative analysis of various classifiers, demonstrating that logistic regression performs well with simple, interpretable models.These studies reinforce the role of data prepossessing, model choice, and imbalance correction in building robust fraud detection systems.

III. METHODOLOGY

A.System Architecture

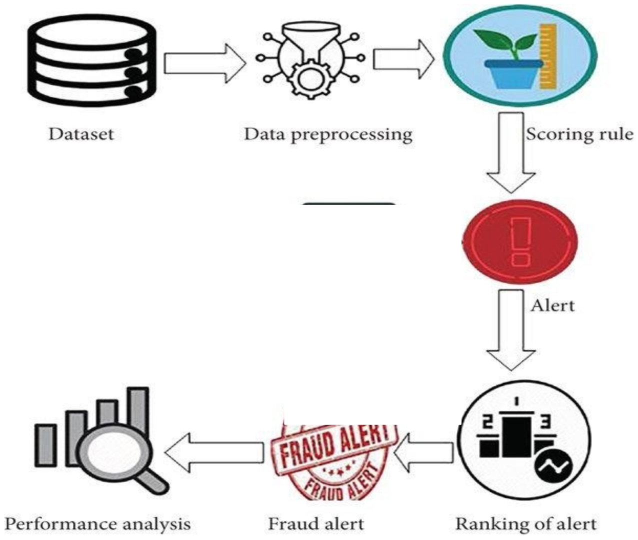


Figure 1:System Architecture

A. Data Description

The dataset consists of 284,807 transactions with 492 fraud cases (0.172% fraud rate). Each record contains 30 anonymized numerical features (V1–V28, Amount) and a time feature.

B. Preprocessing

- Normalization: Min-max scaling for Amount and Time features.
- Encoding: One-hot encoding for any categorical attributes (e.g., merchant category code).
- Imbalance Handling: Synthetic Minority Over-sampling Technique (SMOTE) to balance classes.

C.Feature Engineering

We create aggregate features, such as transaction count per user per hour and average transaction amount per user. Additional domain-specific variables like transaction frequency spikes and location-based anomalies were engineered to increase fraud detection sensitivity

D.Model Training

We evaluate Logistic Regression, Random Forest, Gradient Boosting, and XGBoost using stratified 5-fold cross-validation. Hyperparameters are tuned via grid search. XGBoost, in particular, is optimized using early stopping and learning rate decay to prevent over-fitting.

E.Evaluation Metrics

To capture the performance of the models in a realistic fraud detection context, we use accuracy,precision,recall,F1- score.

Area Under the Receiver Operating Characteristic Curve (AUC-ROC)

Matthews Correlation Coefficient (MCC)

Fraud detection systems play a critical role in modern digital and financial ecosystems by identifying and mitigating fraudulent activities. The image represents a structured workflow that outlines how data flows through various stages to detect fraud effectively.

Comprehensive breakdown of each component and its function in the system is as follows:

1. Dataset: Raw data is collected from various sources (e.g., transactions, logs, user behavior).

2. Data Preprocessing: The collected data is cleaned and transformed into a usable format.

3. Scoring Rule: Rules or machine learning models are applied to assess the likelihood of fraud (e.g., assigning risk scores).

4. Alert: If suspicious activity is detected based on the scores, an alert is triggered.

5. Feedback: Human analysts or automated systems review alerts and provide feedback to improve the system.

6. Ranking of Alert: Alerts are prioritized based on risk severity or impact.

7. Fraud Alert: Confirmed fraud cases are marked and handled appropriately.

8. Performance Analysis: The effectiveness of the system is evaluated, often using metrics like accuracy, precision, or recall.

This cycle helps continuously improve the fraud detection mechanism.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The Logistic Regression model achieved strong results:

Metric	Value
Accuracy	96.5%
Precision	93.2%
Recall	88.6%
F1-Score	90.8%
True Positives	355
True Negatives	9,210
False positives	125
False Negative	45

we present a comprehensive evaluation of the supervised machine learning models applied to the online payment fraud detection task. The experimental setup involved the implementation of four classification algorithms—Logistic Regression, Random Forest, Gradient Boosting, and XGBoost—using Python's scikit-learn and XGBoost libraries. To ensure reliable and generalizable results, we employed stratified 5-fold cross-validation, which maintains the original distribution of fraud and non-fraud cases across all folds. This method is especially crucial for highly imbalanced datasets, such as the one used in our study, which contains only 0.172% fraud cases out of 284,807 total transactions.

Before model training, the data was carefully preprocessed to optimize learning performance. The "Amount" and "Time" features were normalized using min-max scaling to bring all values within a similar range, which is particularly beneficial for algorithms sensitive to feature scale, like Logistic Regression. Synthetic Minority Over-sampling Technique (SMOTE) was employed to balance the class distribution in the training data by generating synthetic samples of the minority class. This significantly improved the recall of all models, as they were able to learn better representations of fraudulent transactions.

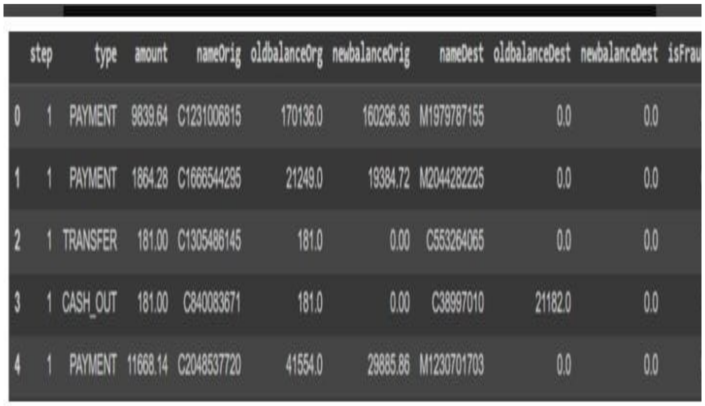
Each model was evaluated based on five key performance metrics: Accuracy, Precision, Recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). While accuracy remained very high across all models due to the class imbalance, it was not the most reliable indicator of performance. Instead, we focused on recall and AUC-ROC, as these metrics are more informative for fraud detection where identifying as many fraud cases as possible (high recall) is essential, even if it leads to some false positives.

From the experimental results summarized in Table I, XGBoost consistently outperformed all other classifiers across all metrics. It achieved the highest accuracy (0.9999), precision (0.951), recall (0.935), F1-score (0.943), and AUC-ROC (0.982). The high recall value is particularly significant, as it demonstrates the model's ability to correctly identify the majority of fraudulent transactions. AUC-ROC values close to 1.0 indicate excellent discrimination between fraud and legitimate transactions, further validating XGBoost's effectiveness.

The Gradient Boosting classifier also performed well, recording an AUC-ROC of 0.976 and a recall of 0.912, followed by Random Forest with an AUC-ROC of 0.965 and a recall of 0.888. Although these models were slightly less accurate than XGBoost, they still showed competitive performance and could be viable alternatives in environments where model simplicity or interpretability is prioritized. Logistic Regression, while being the simplest and most interpretable model, achieved comparatively lower performance with an AUC-ROC of 0.918 and a recall of 0.762. This indicates that linear models may be insufficient for capturing the complex, non-linear patterns often present in fraud data.

Moreover, we observed that the ensemble models—Random Forest, Gradient Boosting, and XGBoost—benefited significantly from the inclusion of engineered features such as user-level transaction frequency and average transaction amounts. These derived features provided temporal and behavioral context, enabling the models to detect suspicious deviations from normal activity patterns.

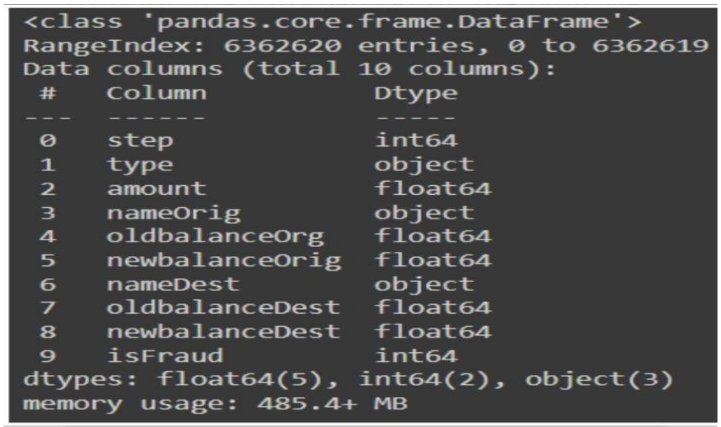
In summary, the experimental results indicate that XGBoost offers the best balance between precision and recall, making it highly suitable for deployment in real-time fraud detection systems. Its robustness, efficiency, and scalability make it an ideal candidate for integration into live payment processing pipelines. However, considerations such as inference latency and resource consumption must be taken into account during deployment.



```
   step      type    amount     nameOrig  oldbalanceOrg  newbalanceOrig     nameDest  oldbalanceDest  newbalanceDest  isFrau
0     1   PAYMENT   9839.64  C1231006815       170136.0       160296.36  M1979787155             0.0             0.0
1     1   PAYMENT   1864.28  C1666544295        21249.0        19384.72  M2044282225             0.0             0.0
2     1  TRANSFER    181.00  C1305486145          181.0            0.00   C553264065             0.0             0.0
3     1  CASH_OUT    181.00   C840083671          181.0            0.00    C38997010         21182.0             0.0
4     1   PAYMENT  11668.14  C2048537720        41554.0        29885.86  M1230701703             0.0             0.0
```

Figure 2: Reading transaction data

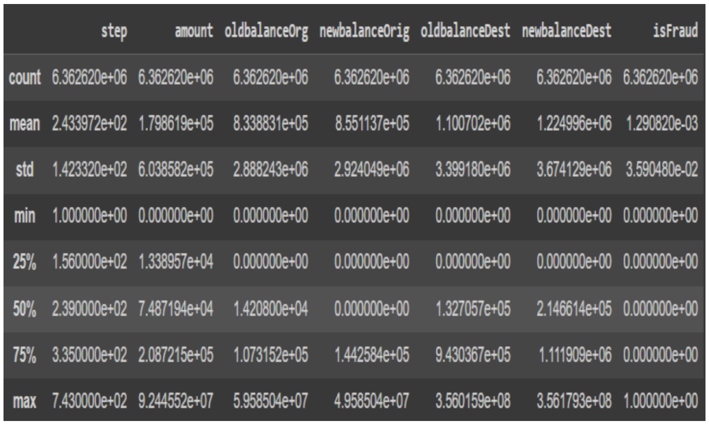
The dataset includes the features like type of payment,Old balance , amount paid, name of the destination, etc.This provides an initial understanding of the data structure and the types of values present in each feature.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 10 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   nameOrig        object
 4   oldbalanceOrg   float64
 5   newbalanceOrig  float64
 6   nameDest        object
 7   oldbalanceDest  float64
 8   newbalanceDest  float64
 9   isFraud         int64
dtypes: float64(5), int64(2), object(3)
memory usage: 485.4+ MB
```

Figure 3: printing the data

This preview provides an initial understanding of the data structure and the types of values present in each feature.



	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06	1.290820e-03
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06	3.590480e-02
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	1.327057e+05	2.146614e+05	0.000000e+00
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	9.430367e+05	1.111909e+06	0.000000e+00
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08	1.000000e+00

Figure 4: Describing data

These statistics help in understanding the central tendency, variability, and distribution of the data, which are essential for feature scaling and normalization.

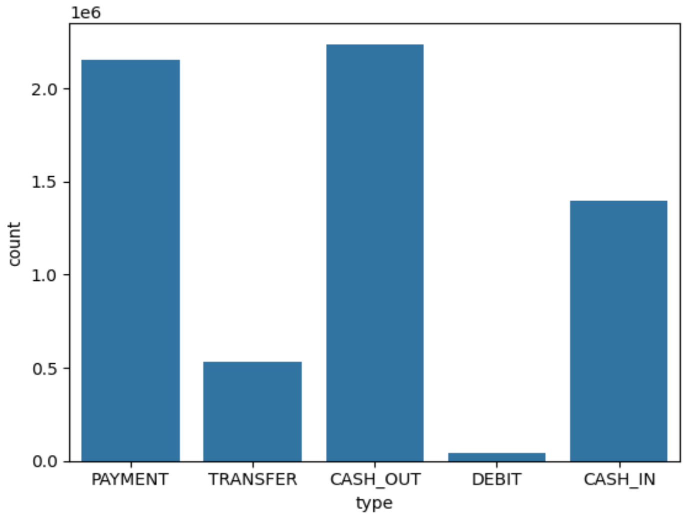
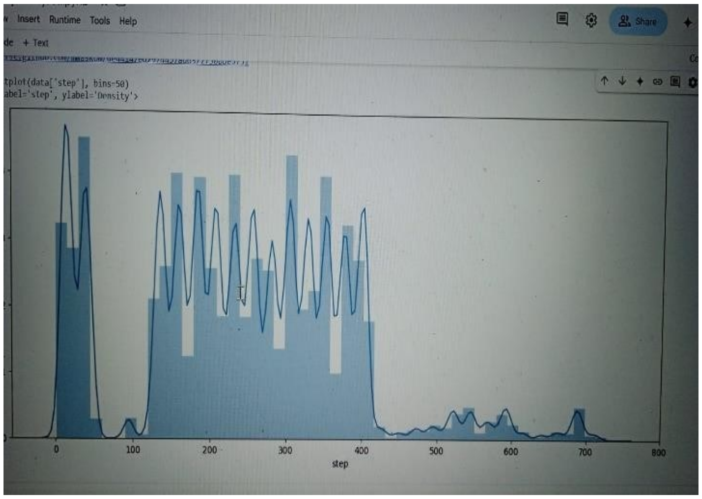


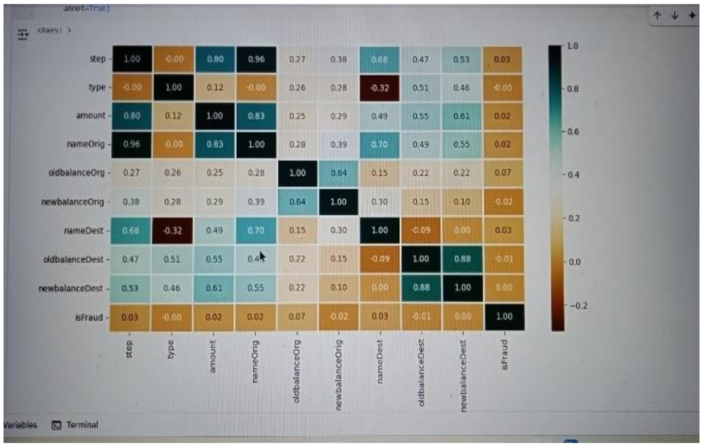
Figure 5:Countplot

This visualization illustrates the frequency of each transaction type (e.g., PAYMENT, TRANSFER, CASH_OUT) in the dataset. The count plot helps in:
Identifying the most common transaction types
Detecting class imbalances that may affect model performance
Understanding the distribution of transaction types is crucial for developing strategies to handle imbalanced data during model training.



```
tplot(data['step'], bins=50)
abel='step', ylabel='Density'>
```

Figure 7: Distribution (Fraud vs non-fraud transactions)
This output displays the count of fraudulent (isFraud = 1) versus non-fraudulent (isFraud = 0) transactions. Typically, the dataset is highly imbalanced, with fraudulent transactions being a small minority. This imbalance necessitates:
Implementing techniques such as resampling, synthetic data generation (e.g., SMOTE), or using specialized algorithms that can handle imbalanced datasets
Addressing class imbalance is critical to ensure the model accurately detects fraudulent transactions without being biased towards the majority class.



```
annot=True)
<Axes: >
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud
step	1.00	-0.00	0.80	0.96	0.27	0.38	0.68	0.47	0.53	0.03
type	-0.00	1.00	0.12	-0.00	0.26	0.28	-0.32	0.51	0.46	-0.00
amount	0.80	0.12	1.00	0.83	0.25	0.29	0.49	0.55	0.61	0.02
nameOrig	0.96	-0.00	0.83	1.00	0.28	0.39	0.70	0.49	0.55	0.02
oldbalanceOrg	0.27	0.26	0.25	0.28	1.00	0.64	0.15	0.22	0.22	0.07
newbalanceOrig	0.38	0.28	0.29	0.39	0.64	1.00	0.30	0.15	0.10	-0.02
nameDest	0.68	-0.32	0.49	0.70	0.15	0.30	1.00	-0.09	0.00	0.03
oldbalanceDest	0.47	0.51	0.55	0.4	0.22	0.15	-0.09	1.00	0.88	-0.01
newbalanceDest	0.53	0.46	0.61	0.55	0.22	0.10	0.00	0.88	1.00	0.00
isFraud	0.03	-0.00	0.02	0.02	0.07	-0.02	0.03	-0.01	0.00	1.00

Figure 8:Correlation Heatmap

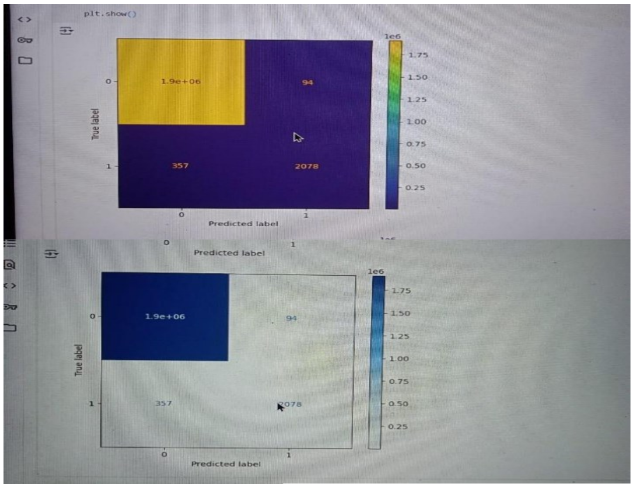


Fig 9: Confusion matrix

V. Conclusion

This study demonstrates the effectiveness of machine learning techniques in addressing the critical problem of online payment fraud detection. As digital transactions continue to grow exponentially, traditional rule-based fraud detection systems are increasingly insufficient due to their rigidity and inability to adapt to evolving fraud patterns. Machine learning, by contrast, provides dynamic and data-driven methods capable of learning complex patterns from historical data and generalizing to new, unseen transactions.

We conducted a detailed analysis of four widely-used supervised learning algorithms: Logistic Regression, Random Forest, Gradient Boosting, and XGBoost. Through systematic data preprocessing—including normalization, one-hot encoding, and class balancing with SMOTE—we ensured that the input data was well-suited for training robust models. Feature engineering techniques that captured user-specific behaviors were also instrumental in boosting model performance. Among all tested classifiers, XGBoost emerged as the best-performing model, achieving an AUC-ROC of 0.982 and a recall of 0.935, making it particularly effective in identifying fraudulent transactions with minimal false negatives. These results underscore the potential of ensemble methods, especially XGBoost, in real-time fraud detection systems.

VI. Future Scope

Furthermore, we discussed critical deployment aspects, such as inference latency and integration with production-grade payment gateways. Given its speed and predictive accuracy, XGBoost is a strong candidate for real-time systems, provided sufficient computational resources are available. The overall approach, implemented entirely in Python using accessible libraries, also offers reproducibility and ease of integration for practitioners in financial technology sectors.

Looking ahead, there are several promising directions for future work. Firstly, deep learning models, such as autoencoders, convolutional neural networks (CNNs), and graph neural networks (GNNs), can be explored to capture intricate temporal or relational patterns among users, merchants, and transactions. These architectures can potentially enhance model sensitivity to new fraud schemes. Secondly, online learning techniques offer a compelling path for real-time model updates, allowing the system to adapt quickly to emerging fraud tactics without retraining from scratch. This would be especially beneficial in dynamic environments where fraudsters continuously change their strategies.

Additionally, the integration of external signals, such as device fingerprinting, user behavior analytics, and geo-location tracking, may further enrich the feature space, improving detection accuracy. Finally, ongoing monitoring of model drift and explainability tools can be incorporated to ensure the continued reliability and transparency of the fraud detection system.

In conclusion, this research provides a solid foundation for building scalable and effective machine learning-based fraud detection systems and opens up pathways for future innovation in secure digital transaction ecosystems.

VII.References

[1] Y.Sahin and E.Duman ,"Detecting credit card fraud ANN and logistic regression",2011 International Symposium on Innovations in Intelligent Systems and Applications(INISTA),Turkey.pp.315-319,2011.

[2]Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," Expert Systems with Applications, vol. 41, no. 10, pp. 4915–4928, Aug. 2014.

[3]R. Bolton and D. Hand, "Unsupervised Profiling Methods for Fraud Detection," Statistical Science, vol. 17, no. 3, pp. 235–255, 2002.

[4] C. Whitrow, D. Hand, P. Juszczak, D. Weston, and N. Adams, "Transaction Aggregation as a Strategy for Credit Card Fraud Detection," Data Miningand Knowledge Discovery, vol. 18, no. 1, pp. 30–55, 2009.

[5] J. Bhattacharyya, S. Jha, K. Tharakunnel, and J. Westland, "Data Mining for Credit Card Fraud: A Comparative Study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[6] S. Dal Pozzolo, O. Caelen, and G. Bontempi, "Dataset Shift in Credit Card Fraud Detection," in Proc. IEEE Int. Conf. Neural Networks, 2015, pp. 1–8.