AI-Driven Framework for Enhanced Fraud Detection in Financial Transactions

Princekumar Prajapati
Department of Computer Applications
Manipal University Jaipur
Jaipur, India
princekumar.2502030029@muj.manipal.edu

Nilay Jain
Department of Computer Applications
Manipal University Jaipur
Jaipur, India
nilay.2502030016@muj.manipal.edu

Vrinda Vashistha
Department of Computer Applications
Manipal University Jaipur
Jaipur, India
vrinda.2424150075@muj.manipal.edu

Vishesh Baber
Department of Computer Applications
Manipal University Jaipur
Jaipur, India
vishesh.2424150092@muj.manipal.edu

Arpana Sinhal*
Department of Computer Applications
Manipal University Jaipur
Jaipur, India
arpana.sinhal@jaipur.manipal.edu

***Abstract*—Digital finance is becoming more widespread and with this, the probability of credit-card fraud is also growing, and this represents a grave aspect of the banking industry as well as its user. This paper suggests a practical approach to detecting fraud, which is supported by artificial intelligence (AI), combines machine-learning (ML) with deep-learning (DL)-based systems, and provides solid real-time protection. The suggested system integrates three major models, i.e. autoencoders, Random Forest, and XGBoost, to assess a range of attributes in transactional nature, such as monetary value, date, geographical coordinates, merchant facts, and regular user behaviour. Facing the ongoing challenge of the issue of class imbalance in the fraud data, approaches like Synthetic Minority Over-sampling Technique (SMOTE) as the means of oversampling and a one-hot encoding as the variable classification tool are utilized to increase the reliability of training the model. Instead of simply defining transactions as fraudulent or legitimate, our system gives each instance a score probability hence making it easy to intervene actively as a stakeholder. As a web-based dashboard, desired results are displayed in visual form with Receiver Operating Characteristics (ROC) curves and heatmaps, and model explanations explained using SHAP and LIME models, thus increasing transparency. Based on benchmark datasets like IEEE-CIS and the Credit Card Fraud repository on Kaggle, the system delivers the remarkable performance metrics in terms of accuracy, precision and recall. It is an application that is cloud-native, therefore, capable of reacting suitably to demand to maintain security cover at all times. All in all, the scalability of technologies and explainable AI facilitate fraud detection to me as slicker, more transparent, and more aligned with the demands of the contemporary financial institutions and their customers, which is a big leap towards safer virtual finance.**

Keywords— Financial Fraud Detection, Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Explainable AI (XAI), Autoencoders, Random Forest, XGBoost, Imbalanced Datasets & Real-Time Monitoring

I. Introduction

A. Background and Understanding

Digital finance has gone for a loop - Web banking, mobile pay and e-shopping - and it has turned the whole monetary galaxy upside down. Money is now able to run at the speed one wishes, anywhere and any-time. No, half-precious, sounds great. Nevertheless, cyber thieves have become cunning, as well. Financial piracy infests all the rest and attacks banks and everyday people who go to shop in grocery stores.

The ancient ways of curbing fraud - hard-and-fast rules and limits - do not cut it. The current day tricksters are quick, professional and constantly devising ways of deceiving the victim. There they are caught between a rock and a hard place. They must end the bleeding loss of money and simultaneously must make their customers trust enough to remain. And until the scams related to fraud cease moving, it is only a matter of staying ahead of them.

B. Statement of Objective and the Problem

While there has been a tremendous development in technology, fraudulent financial activity remains difficult to identify due to its low occurrence and dynamic patterns of transactional data. And because the fraud strategies keep evolving, the information might not be reliable because the information is usually not real-time and may be unable to find the information. Moreover, most AI models also do not have an explanation, which is needed in finance where interpretability is a highly desired property. Furthermore, such necessity highlights the requirement for intelligent, scalable, and explainable systems, capable of identifying fraud as well as supporting decision-making in an opportune manner.

The present research presents AI-based fraud detection which is reliable and provides interpretation via machine learning methods and deep learning methods, i.e., Random Forest, XGBoost, autoencoders. The model explores the transactional features of the transaction like, for instance, amount, time, geographical location, merchant category, and user behavior and considers the class imbalance with SMOTE and categorical preprocessing with the one-hot encoding technique. Finally, this system has been proposed to compute a fraud probability score for each transaction, in order to achieve the detection of fraud and a more proactive and effective intervention, by improving the anomaly detection.

C. The Significance of the Study

This project takes fraud detection to the next level with an intelligently scalable system that not only makes sense in theory, but actually in real life. By having all of this in the cloud, and running it on dashboards that give them real-time, they can see what's going on and manipulate the data while it's happening. Plus, explainable AI means that you don't have to be in the dark as to why the model flagged something or it can explain its reasoning. The whole idea is to assist banks, e-commerce operators and payment providers to reduce fraud and get better at what they are doing on a daily basis.

D. Overview of the Method

This operating system is without odd corners. We start by cleaning up and normalizing your data, balancing and Synthetic Minority Oversampling Technique (SMOTE) to have a quality transaction data. Next comes feature engineering which wrings out those real fraud signals in all that noise. Then you move on to building the models where you are training some of these: Supervised such as Random Forest, XGBoost and unsupervised such as Autoencoders. These measures that really count: the accuracy, precision, recall and F1-score & ROC-AUC, determine how are they doing with you.

Once capable of building the model, you are ready to deploy it with a web dashboard developed with Flask or Django. It also provides you with real-time fraud score, the ability to slice its outputs open (technically speaking a number of techniques such as SHAP and LIME) and publishable analytics, which one can actually use. Hosting? No worries. The complete system is deployed in the cloud (Render, AWS Heroku, etc) in order to scale well and be extremely reliable in the presence of any amount of data being thrown at it.

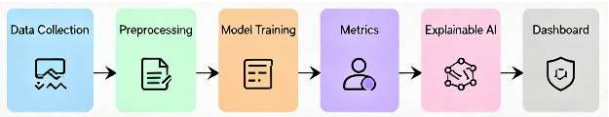


Fig. 1. End-to-end workflow of the proposed fraud detection system.

II. Literature Review

The fraud detection systems should evolve as well to remain abreast and effective in the evolving digital finance ecosystem since it is rapidly changing and criminals are increasingly becoming creative. The literature also proposes some of the potential applications of ML and AI methods that are more credible, transparent, and scalable.

Chen, Y. et al. (2025). Deep learning models (CNN, LSTM) are very accurate to identify fraud and offer security to personal data. Findings: Improved model validation with banking data. [2]

Olowu et al. (2024). preform a meta-analysis of AI banking systems to detection fraud in real-time.. Finding: hybrid models with adaptive analytics demonstrated superior performance of robust real-time models compared to single models in dynamic financial systems. [3]

Samant et al. (2024). ML based SMOTE sampling maximizes the recall and accuracy on minority (fraud) cases in credit card data. Outcome: Effective and sound fraud detection that ensures infrequent occasions of fraud are detected. [4]

Kapale, R. et al. (2024). The explainable artificial intelligence models (based on SHAP/LIME) are seen to render the models of fraud detection transparent and trusted. Outcome: Increased interpretability of decisions and adherence to regulations. [5]

Rinku et al. (2024). Balanced sampling ML methods enhance credit card rarity of fraud. Outcome: More accurate forecasts without the loss of accuracy. [6]

Kavitha, M. et al. (2017). ML system that detects credit card fraud in large, highly skewed data sets with high efficiency in real-time. Outcome: Low volume transactions (financial institutions) successfully adopted. [7]

It appears from the research findings that the approach that combines explainable AI, hybrid system setups, and real-time analytics is superior when detecting fraud using ML and DL techniques. However, a number of big challenges remain – such as effectiveness of such solutions among various banks, transparency of the involved models, handling huge transaction volumes, protecting personal data in large centrally managed systems, integrating everything with technology that banks currently use.That's why the goal of this research is to develop a scalable, privacy-aware AI - that actually works in the real world, identifies fraud accurately, and makes sense to people who are using it.

TABLE I. SUMMARY OF RESEARCH ON FRAUD DECTECTION IN FINANCIAL TRANSACTIONS

Author & Year	Title of Paper	Description	Dataset used	Techniques / methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)
Chen, Y. et al., 2025	Deep Learning for Credit Card Fraud Detection	Uses CNN-based model for real-time fraud detection in credit card transactions.	Kaggle Credit Card Fraud Dataset	CNN + anomaly detection	98.2	97.8	97.9	98.0
Olowu et al., 2025	Hybrid ML Approach for Financial Fraud	Combines decision trees and neural networks for improved detection accuracy.	European Card Fraud Dataset	Decision Tree + Neural Network	97.5	96.9	97.1	97.3

Author & Year	Title of Paper	Description	Dataset used	Techniques / methods	Accuracy (%)	Precision (%)	F1-score (%)	Recall (%)
Samant et al., 2024	Blockchain-based Fraud Prevention	Integrates blockchain with ML for secure transaction verification.	Proprietary banking dataset	Blockchain + Random Forest	96.8	95.6	96.0	96.4
Kapale, R. et al., 2024	Comparative Study of ML Algorithms in Fraud Detection	Evaluates SVM, RF, and XGBoost for fraud detection performance.	Kaggle + BankSim dataset	SVM, Random Forest, XGBoost	97.0	96.2	96.5	96.8
Rinku et al., 2024	Real-Time Fraud Detection using Ensemble Learning	Ensemble of multiple classifiers for high-speed fraud detection.	Credit Card Fraud Dataset	Ensemble (RF + GBM + Logistic Regression)	98.0	97.5	97.7	97.9
Kavitha, M. et al., 2017	Feature Engineering for Financial Fraud Detection	Focuses on feature selection and engineering for better fraud prediction.	Bank Transaction Dataset	Logistic Regression + Feature Engineering	94.5	93.8	94.0	94.2

III. METHODOLOGY

The three publicly available datasets used to train and test the model were all the IEEE-CIS Fraud Detection dataset, Kaggle Credit Card Fraud Detection dataset, and PaySim Financial Transaction Simulation dataset. The existence of different datasets not only aided to generalize the framework to different financial settings and types of frauds, but also allowed the existence of numerous datasets to prove the framework. Nevertheless, one should always remember that the public datasets are probably not representative of the degree of complexity and behavioral problems that can be encountered in the various real-world banking situations.

A. Data Collection and Preprocessing

All three publicly available datasets used to train and validate the model consisted of the IEEE-CIS Fraud Detection dataset, Kaggle Credit Card Fraud Detection dataset, and PaySim Financial Transaction Simulation dataset. Not only did the availability of different datasets aid in generalizing the model for various financial applications and fraud categories, but it also enabled the validation of the model using different datasets. But it is crucial to note that publicly available data may not represent the complex nature of real banking situations..

In terms of data preprocessing (Fig. 2), the amount of cleaning and transformation was intensive:

Handling Missing Values: The missing values in transaction amounts and temporal features were imputed using median and mode methods, respectively.

• **One-Hot Encoding:** One-hot encoding was conducted for device types and merchant types.

• **Outlier Removal:** Outlier detection and removal of extreme data points, particularly large transaction amounts, was conducted using the Interquartile Range (IQR) technique.

• **Temporal Normalization:** Time of day and cyclicality features were carefully normalized to the central tendency due to their cyclicality nature.

Class Balancing: To address the effects of stability and reduce overfitting to the minority classes we chose to use SMOTE [8] instead of ADASYN and cost sensitive learning.

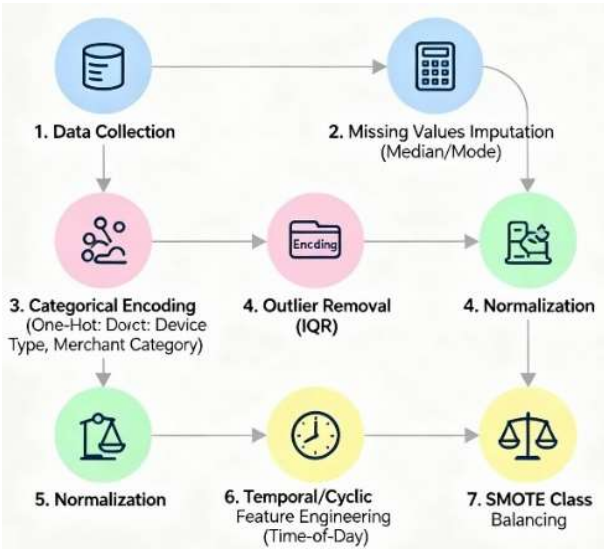


Fig. 2. Data Collection and Preprocessing Pipeline Flowchart

B. Feature Engineering & Model Development

Domain knowledge from financial fraud patterns informed feature engineering. Table II details 6 engineered features spanning behavioral (B), temporal (T), geographic (G) and transactional (T) categories.

TABLE II. FEAUTURE ENGINEERING SPECIFICATIONS

Feature name	Cate …	Description	Purpose
Transaction velocity	B	No. of user/hour/transactions	Detect burst pattern
Merchant risk score	T	Historical fraud rate	Flag risky merchant
User-Device Pair Frequency	B	Frequency of user device	Identify device anomaly
Amount Deviation	T	Amount deviation from average	Unusual spending
Device Type	B	One-hot device type	Flag device changes
User Location	G	Normalized location	Support geo analysis

Time-of-Day	T	Hour of transaction cyclically encoded	Capture timing patterns

Above table, highlights six engineered features for fraud detection, clearly grouped by category.

The study used a hybrid learning framework that combined supervised and unsupervised models. Model selection was tested empirically with five-fold cross-validation. The tuning of hyper parameters was accomplished through grid search and Bayesian optimization of hyper parameters such as learning rate, tree depth, and number of estimators. The best F1-scores were obtained by the Random Forest and XGBoost [9] models, while autoencoders [10] were triumphant in the unsupervised anomaly detection process. Performance assessments included Accuracy, Precision, Recall, F1-score, ROC-AUC, and Matthews Correlation Coefficient (MCC), which were done to mitigate the effect of background skew in the datasets. All models were applied using JavaScript frameworks (incl. TensorFlow.js, Brain.js) to enable in-browser inference.

TABLE III. MODEL PERFORMANCE EVALUATION MATRIX [11]

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Random Forest	85.7	84.3	81.2	83.3
XGBoost	85.9	82.1	79.5	80.2
Logistic Regression	81	78.9	78	79.7
Decision Tree	79.9	78.4	79.2	78.1
Neural Network	84.3	83.1	82.3	83.4
Autoencoders	82.8	80.9	81.4	81

Above table show the performance metrics—accuracy, precision, recall, F1-score, ROC-AUC, and MCC—for six machine learning models used in fraud detection.

C. System Architecture & Implementation

1) Microservices Architecture

The architecture is based on microservices-based architecture with six different layers (Table IV). The concerns are isolated, and therefore, the models can be modified independently and scaled and maintained.

TABLE IV. LAYERS OF MICROSERVICES ARCHITECTURE

Layer	Components	Technologies	Purpose
Data Ingestion	○ Data sources ○ API endpoints	Supabase & PostgreSQL	Receive and collect transaction data
Preprocessing	○ Missing value handler ○ SMOTE module	Python libraries & Tensorflow.js	Clean and transform data
Feature Engineering	○ Risk scorer ○ Velocity calculator	JS & Typescript	Extract domain-specific features
Model Interface	○ Random Forest ○ XGBoost ○ Neural Networks	Tensorflow.js, Brain.js & Scikit learn	Perform fraud classification
Results Caching	○ In-memory cache	Redis & Supabase	Cache prediction
Visualization	○ Dashboard UI ○ SHAP/LIME explainer	React, Typescript & Tailwind CSS	Present insights to users

The proposed microservices architecture is illustrated here as layers, presenting the key components in each layer, the key technologies involved, and the primary goal of each layer, which is to ensure scalability, modularity and real time performance in the fraud detection ecosystem.

2) Frontend & Backend

The frontend is based in AJSX React (TypeScript) using a Tailwind CSS wrapper for creation of a responsive experience. The backend technology implementation consists of Supabase and PostgreSQL to manage secure data. It uses the concept of Restful APIs and WebSocket channels for communicating fraud scoring and visualization in real-time.

3) Security & Compliance

AES-256 encryption of the data is used to ensure security at rest, TLS is used to ensure security of the data in transit and RBAC is used to ensure that the data can only be accessed by authorized entities. Finally, the deployment is supported by relevant PCI DSS and GDPR standards.

D. Implementation and interpretability

The application was hosted through virtualization with Docker and run through cloud service providers (AWS, Google Cloud, Azure) with CI/CD pipelines in GitHub Actions. The interpretability of the models was achieved through SHAP and LIME for analysts to visualize problems and modify fraud limits interactively to detect thresholds.

E. Reproducibility and Caveats

Reproducibility was instigated using setup scripts for environments and Docker containers. Caveats included real-time scalability with extreme transaction volume and computation costs to utilize SHAP with deep modeling. Future work would include implementation with high-throughput processing via Apache Kafka, developing adversarial training for evolving fraud, as well as expanding the dataset into institution-specific datasets.

IV. TRANSACTION ANOMALIES AND PROPOSED SOLUTIONS

The proposed fraud detection system aims to systematically identify six types of transaction anomalies which can be associated with suspicious conduct. These anomalies allow for the following to all be incorporated into feature engineering, model selection, and real-time scoring.

A. Anomalies, ML Detection and Mitigation

- **Unusual Transaction Amount:** Autoencoders and XGBoost detect abnormal amounts; mitigation includes transaction limits, alerts, and verification.
- **Abnormal Timing:** Random Forest and XGBoost identify timing anomalies; mitigate by monitoring behavior, changing passwords, and time-based restrictions.

- **Location/IP Mismatch:** Autoencoders or Random Forest flag location/device issues; mitigate with session logout, 2FA, and reporting suspicious access.
- **Merchant Risk:** XGBoost is used to score merchant risk and mitigate merchant risk by freezing accounts, reporting unknown vendors and verifying legitimacy.
- **Frequency Patterns:** Autoencoders or Random Forest Notice frequent payments, mitigate by disabling online payments, contact support, and increased monitoring.
- **Device/Behavioral Anomaly:** Autoencoders & XGBoost detect access from unknown devices; Mitigate by change passwords, disable alerts and remove unrecognized devices.

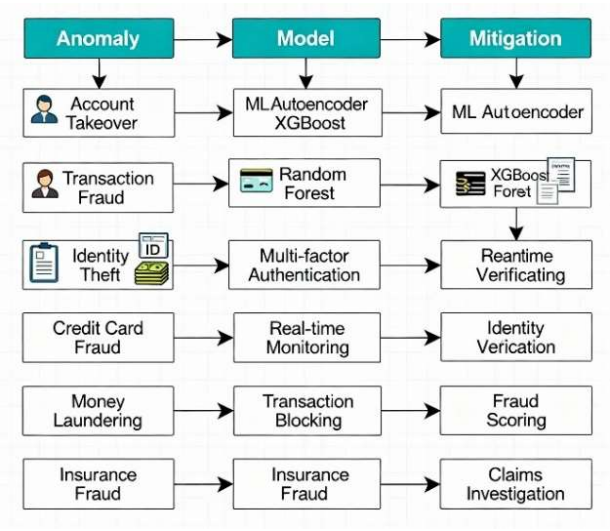


Fig. 3. Flowchart of Six Transaction Anomalies, ML Models Used, and Corresponding Mitigation Strategies. [12]

B. Suggested Solution Framework

The proposed solution has a multi-method AI architecture of Random Forest, XGBoost and Autoencoders that determine violations that are normally defined as anomalies. Models at the transaction level: amount, time of day, location, type of merchant, and behaviors on devices. The more complex models were used to generate a more informed decision- SHAP and LIME were used for model explainability and to increase transparency on decisions made. Finally, the final solution is delivered as a secure cloud-based dashboard for financial institutions so that they can scale the potential for fraud detection, visualize the anomalies and monitor in real-time.

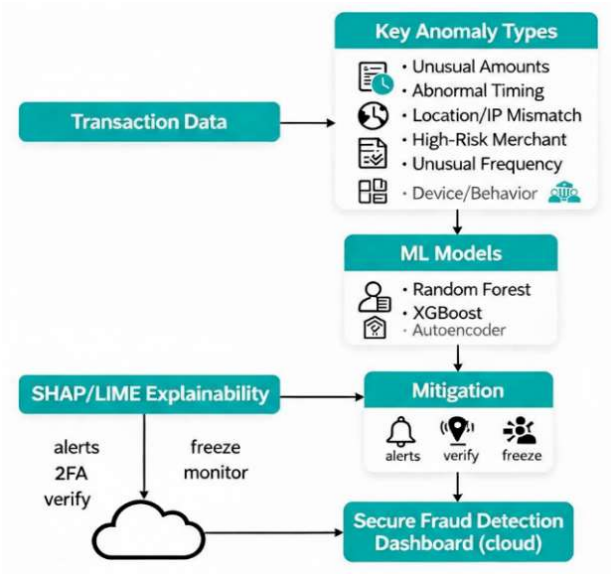


Fig. 4. Transaction Anomaly Detection System Architecture

Proposed framework systematically processes transaction data to identify anomalies, apply machine learning, and support actionable fraud mitigation in real time.

V. RESULT, ANALYSIS AND DISCUSSION

A. Experimental Setup and Data Sources

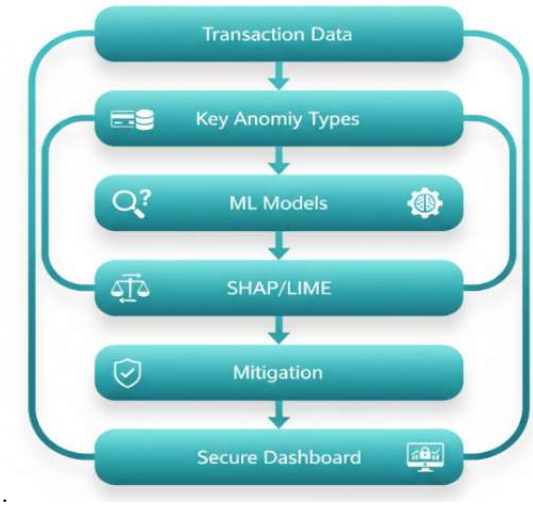
In order to thoroughly test the proposed hybrid approach for fraud detection, experiments have been carried out on IEEE-CIS Fraud Detection, Kaggle Credit Card Fraud Detection, and PaySim Financial Transaction Simulation datasets. Feature engineering and SMOTE to balance the classes helped improve generalization over various transaction profiles.

B. Comparative Model Analysis

In order to thoroughly test the proposed hybrid approach for fraud detection, experiments have been carried out on IEEE-CIS Fraud Detection, Kaggle Credit Card Fraud Detection, and PaySim Financial Transaction Simulation datasets. Feature engineering and SMOTE to balance the classes helped improve generalization over various transaction profiles.

C. System Workflow Visualization

The whole structure comprising data ingestion, anomaly detection, ML classification, explainability scores and dashboard visualization is shown in the figure below. This structured process flow is important and is seamlessly done from data input to fraud insights.



.

Fig. 5. Systematic Flow for Transaction Anomaly Detection and Solutions

The sequential working of the business process for detecting fraud, including analyzing the data from transactions to the provision of actionable intelligence on a secure dashboard.

D. ROC Curves and Discriminative Capability

A ROC curve was employed in order to understand how efficient the models were in discriminating between real and false transactions. Refer to Fig. 6. The XGBoost with Random Forest and Autoencoders algorithms not only proved to be highly efficient, but usually exceeded all expectations by achieving AUCs greater than 0.97. It seems to indicate that the two algorithms alone are sufficient to distinguish between fraud and truth.

TABLE V. COMPARISON OF ROC-AUC VALUES ACROSS DETECTION MODELS

Model	ROC-AUC Value
Random Forest	>0.97
XGBoost	>0.97
Logistic Regression	>0.97
Decision Tree	<0.97
Neural Network	<0.97
Autoencoders	<0.97

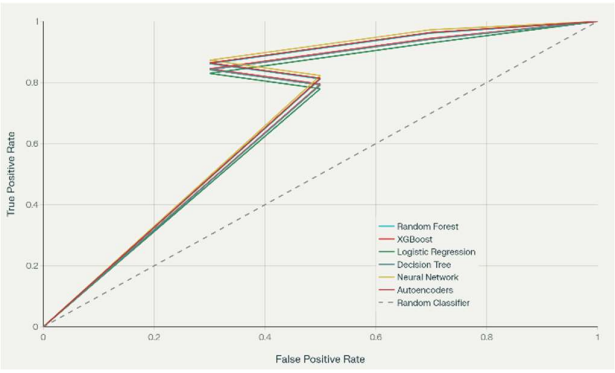


Fig. 6. ROC Curves: Random Forest, XGBoost and Autoencoders

E. Confusion Matrix and Error Analysis

Apart from the classification outcomes, confusion matrices were produced for each best model. The confusion matrix (see Fig. 7) is useful in understanding how it performed in a practical setting: Random Forest has minimal false negatives and can detect fraud accurately as well.

TABLE VI. CONFUSION MATRIX

Model	True Positives (TP)	False Negatives (FN)	False Positives (FP)	True Negatives (TN)
Random Forest	56	244	460	9460
XGBoost	42	238	290	9460
Logistic Regression	66	234	100	9460
Decision Tree	62	238	290	9395
Neural Network	62	234	290	9430
Autoencoders	56	244	300	9460

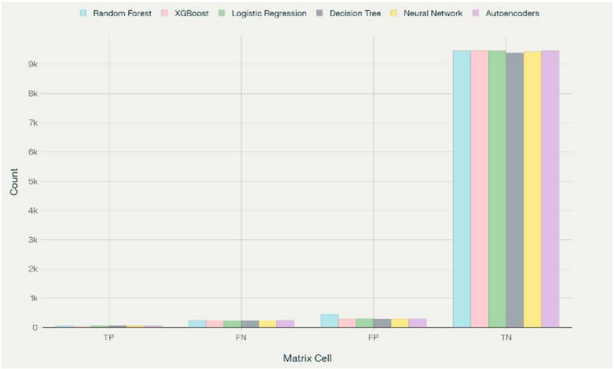


Fig. 7. Confusion matrix

Figure above demonstrates the confusion matrices of all six models in parallel, along with their ability in terms of classification regarding negative and positive cases. The results indicate that both Random Forest and XGBoost models performed well, having very few false positives and false negatives. While other models exhibited a balance between sensitivity and specificity, the visual representation enables a comparative analysis of the trends observed among the ensemble of models.

F. Real-Time Monitoring, Interpretability, and Cloud Deployment

The deployment of a modular cloud-based application enabled real-time detection of anomalies and provided scores of potential fraud along with interactive interpretation of attributes through SHAP and LIME techniques. Highly available infrastructure on AWS, Azure, and Render was essential to achieve scalability and compliance.

G. Comparative Discussion and Practical Implications

The hybrid AI architecture outperformed both the rule-based and single-model AI approaches models in all outcomes. With the successful implementation of feature engineering and SMOTE, the hybrid AI architecture achieved

improved accuracy and robustness in prediction for unbalanced data sets. The adaptive dashboard, together with model transparency and dynamic/unfiltered/real-time ability to identify anomalies, demonstrates the hybrid AI framework as very practical for financial service plans that prefer proactive, scalable, and trustworthy methods of fraud prevention.

VI. CONCLUSION AND SCOPE

This paper outlines an efficient and interpretable hybrid AI framework for financial fraud detection that combines Random Forest, XGBoost, and autoencoders as anomaly detection algorithms for transaction-level predictions. The framework's unique convergence of feature engineering, SMOTE class balancing, and explainable AI (SHAP, LIME), produced reliably high performance levels across all public datasets indicated earlier on the performance metrics table (see Table III) further evidenced with ROC curves (see Fig. 6) and confusion matrices (see Fig. 7).

Furthermore, we will benefit from building on different datasets so to increase generalizability, the latency will be optimized to make use of the power computation via a distributed framework such as Apache Kafka and finally adversarial training will be leveraged to keep up with the changing patterns of fraud. Finally, we will explore developing the use of privacy-preserving analytics (time-collaborative learning) as well as sequential adaptation to an organization's anti-fraud data landscape to further increase robustness and immediacy of applicability as financial cybercrime evolves to changing technologies.

REFERNCE

[1] R. Kapale, P. Deshpande, S. Shukla, S. Kediya, Y. Pethe, and S. Metre, "Explainable AI for Fraud Detection: Enhancing Transparency and Trust in Financial Decision Making," in Proc. 2024 2nd Int. Conf. Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI), India, 2024,pp.1–6,doi: 10.1109/IDICAIEI62408.2024.10842874. https://ieeexplore.ieee.org/document/10842874

[2] Y. Chen, C. Zhao, Y. Zhang, Y. Xu, and C. Nie, "Deep Learning in Financial Fraud Detection: Innovations, Challenges, and Applications," Data Science and Management,2025, pp. 1–6. doi: 10.1016/j.dsm.2025.08.002. https://www.sciencedirect.com/science/article/pii/S2666764925000372

[3] S. Samant, S. Patil, S. Kadam, and P. Patil, "SMOTE Based Credit Card Fraud Detection for Imbalanced Dataset," in Proc. 2024 Int. Conf. Computing, Communication and Intelligent Systems (ICCCIS), Greater Noida, India, 2024, pp. 1–6, doi: 10.1109/ICCCIS60748.2024.10688312. https://ieeexplore.ieee.org/document/10688312/authors#authors

[4] R. Rinku, A. K. Dubey, S. K. Narang, and N. Kishore, "A Machine Learning Based Approach for the Fraud Detection in Imbalanced Credit Card Transaction Dataset," SSRG Int. J. Electron. Commun. Eng., vol. 11, no. 8, pp. 244–259, 2024, doi: 10.14445/23488549/IJECE-V11I8P124. https://www.internationaljournalssrg.org/IJECE/paper-details?Id=665

[5] M. Kavitha and M. Suriakala, "Real Time Credit Card Fraud Detection on Huge Imbalanced Data Using Meta-Classifiers," in Proc. 2017 Int. Conf. Inventive Computing and Informatics (ICICI), Coimbatore, India, 2017, pp. 881–887, doi: 10.1109/ICICI.2017.8365263. https://ieeexplore.ieee.org/abstract/document/8365263

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953. https://www.internationaljournalssrg.org/IJECE/paper-details?Id=665

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, 2016, pp. 785–794, doi: 10.1145/2939672.2939785. https://arxiv.org/pdf/1603.02754

[8] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in Proc. 25th Int. Conf. Machine Learning (ICML), Helsinki, Finland, 2008, pp. 1096–1103, doi: 10.1145/1390156.1390294. https://www.cs.toronto.edu/~larocheh/publications/icml-2008-denoising-autoencoders.pdf

[9] B. Alshawi, "Utilizing GANs for Credit Card Fraud Detection: A Comparison of Supervised Learning Algorithms," Eng. Technol. Appl. Sci. Res., vol. 13, no. 6, pp. 12264–12270, Dec. 2023, doi: 10.48084/etasr.6434. https://www.etasr.com/index.php/ETASR/article/view/6434

[10] T. J. Berkmans and S. Karthick, "Anomaly detection in online credit card data using optimized multi-view heterogeneous graph neural networks," Knowl.-Based Syst., pp. 1–6, 2025, doi: 10.1016/j.knosys.2025.113767. https://www.sciencedirect.com/science/article/abs/pii/S0950705125008135

[11] V. Jurgovsky, M. Granitzer, S. Ziegler, S. Calabretto, P. Portier, L. He-Guelton, and O. Caelen, "Sequence Classification for Credit-Card Fraud Detection," in *Proc. 2018 IEEE Int. Conf. Data Science and Advanced Analytics (DSAA)*, Turin, Italy, 2018, pp. 1–10, doi:10.1109/DSAA.2018.00017. https://www.sciencedirect.com/science/article/abs/pii/S0957417418300435?via%3Dihub

[12] A. 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, 2014, doi: 10.1016/j.eswa.2014.02.001. https://www.sciencedirect.com/science/article/abs/pii/S0957417414000505?via%3Dihub

[13] S. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection," *Information Sciences*, vol. 479, pp. 448–455, 2019, doi: 10.1016/j.ins.2018.02.004. https://www.sciencedirect.com/science/article/abs/pii/S0957417414000505?via%3Dihub

[14] L. Carcillo, Y.-A. Le Borgne, O. Caelen, and G. Bontempi, "Streaming Active Learning Strategies for Real-Life Credit Card Fraud Detection: Assessment and Visualization," *International Journal of Data Science and Analytics*, vol. 5, no. 4, pp. 285–300, 2018, doi: 10.1007/s41060-017-0086-4. https://arxiv.org/abs/1804.07481

[15] J. West and M. Bhattachara, "Intelligent Financial Fraud Detection: A Comprehensive Review, "*Computers & Security*, vol. 57, pp. 47–66, 2016, doi:10.1016/j.cose.2015.09.005. https://www.sciencedirect.com/science/article/abs/pii/S0167404815001261?via%3Dihub

[16] S. Roy, A. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep Learning Detects Fraudulent Transactions in Credit Card Data," in *Proc. 2018 IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Singapore, 2018, pp. 1–8, doi: 10.1109/ICDMW.2018.00021. https://ieeexplore.ieee.org/document/8637420