

Explainable Personalized Behavioural Fraud Detection Using Deviation-Based Feature Engineering and Deep Learning

Shivam
IIMT College of Engineering
Greater Noida, India
shivamsinghraje@gmail.com

Shaurya Rakesh
IIMT College of Engineering
Greater Noida, India
rakeshshaurya571@gmail.com

Rakhi Singh
IIMT College of Engineering
Greater Noida, India
research.rs2k11@gmail.com

Abstract— *Financial fraud continues to be a major problem in digital payment systems. This work introduces a personalized system for detecting fraud that uses behaviour patterns, builds features by looking at how they differ from usual activity, and applies a multilayer perceptron along with SHapley Additive exPlanations to help explain the results clearly. Using the PaySim dataset, user profiles and normalized deviation features help detect unusual behaviour patterns. The model has a high accuracy of 99.5%, good precision of 56.4%, strong recall of 95.8%, a fair F1-score of 71.5%, and a very high area under the curve of 99.8%, showing it can detect fraud accurately and clearly explain its findings.*

Keywords— *Behavioural deviation, deep neural network, fraud detection, personalized user profiling*

I. INTRODUCTION

The quick growth of digital financial services has led to a big increase in fraudulent activities, which creates serious economic and reputational problems for financial institutions around the world. Global financial crime reports show that billions of dollars are lost each year due to payment fraud, which means there's a need for strong, flexible, and smart ways to detect these frauds. Traditional rule-based systems, although they worked well for identifying known types of fraud in the past, are not flexible enough to spot new and more complex forms of fraud that are constantly developing. Machine learning and deep learning methods have become very popular as alternatives, but many existing systems evaluate transactions in isolation, ignoring the temporal and behavioural context within which each transaction occurs.

A major limitation of conventional fraud detection models is that they use raw transactional features such as transaction amount, account balances, and timestamps without relating these values against an individual user's historical behaviour. A transaction with a certain amount of money might be completely normal for one person but seen as a big problem for another. This gap in personalization is a major problem that reduces detection precision and elevates false positive rates, ultimately eroding user trust and operational efficiency.

Behavioural modelling has become a promising approach to deal with this limitation. By constructing user-specific profiles that capture habitual transaction patterns, temporal rhythms, and balance dynamics, fraud detection systems can evaluate each transaction relative to established individual norms rather than global statistical thresholds. If a transaction is very different from how a user usually behaves, it sends an important message that needs more attention. This approach focuses on looking at unusual behaviour rather than just the total amount of a transaction, which helps spot more subtle and personalized fraud patterns more effectively.

Simultaneously, the adoption of deep learning architectures, especially multilayer perceptron networks, has shown better results in modelling complex nonlinear relationships among financial features. However, the difficulty in understanding how deep neural networks work has prompted regulatory and ethical concerns regarding automated decision-making in high-stakes financial domains. Explainability tools like SHapley Additive exPlanations help with this by showing how each input factor contributes to the prediction result, making it easier for people to understand and reason about each fraud decision.

While previous studies have explored behavioural fraud detection and explainable artificial intelligence separately, but this work combines deviation-based personalized behaviour modelling with explainable fraud detection into one single system. This paper makes several key contributions. First, it proposes a structured method for behavioural feature engineering that creates personalized user profiles based on past transaction history. Next, it develops normalized deviation features to capture how much a given transaction differs from a user's typical behaviour. The study also trains a deep neural network on this enriched set of features, using class-weighting techniques to deal with imbalanced data. In addition, SHapley Additive exPlanations are used to make the model's predictions at the transaction level more interpretable. Finally, experiments on the PaySim dataset show that the proposed method performs better than existing baseline approaches.

II. LITERATURE SURVEY

Financial fraud detection has been widely researched in many different areas, including how people behave, using groups of learning methods, advanced learning techniques, and AI that can explain its decisions. This part looks at the most important works related to the topic and finds out what issues the new framework aims to solve.

Early foundational work by Delamaire et al. [9] provided a detailed look at methods used to detect credit card fraud, and they found that looking only at features from individual transactions isn't enough to accurately identify fraudulent activity. Later, Whitrow et al. [11] showed that grouping transactions together helps in finding things more accurately by looking at how people behave over time, which set the stage for methods that create profiles based on individual users. Dal Pozzolo et al. [8] shared real-life experiences from fraud detection systems, pointing out important issues like concept drift and class imbalance that still impact current detection methods.

Many studies have looked into using groups of different methods or mixing different techniques for classification.

Zareapoor et al. [14] used bagging with different classifiers to detect credit card fraud. They showed that using multiple classifiers together worked better than using just one, because it reduced the differences in how each classifier made predictions. Nguyen et al. [15] created a mixed model that uses both gradient boosting and deep neural networks, which worked well on standard test data, but they didn't include personal behaviour information in their approach. Islam et al. [12] did a detailed study on using artificial intelligence for detecting fraud in financial institutions. They found that deep learning methods work better than older techniques. However, they also pointed out that explaining how these AI systems make decisions is still a big challenge that needs to be solved.

Behavioural modelling has become a promising approach for enhancing the ability to detect fraud. Sadgali et al. [1] created a way to evaluate human behaviour by giving risk scores based on how cardholders use their cards. They showed that using this behavioural information helps to be more accurate in identifying risks. However, the method uses manually created scoring rules and does not provide clear explanations. Zhang et al. [2] created a model called the hierarchical behaviour-knowledge space. This model groups user actions into different levels of detail. While this approach works well in spotting things, it uses a lot of computer power and doesn't explain its decisions clearly. Sondinti et al. [3] looked into using federated learning and privacy-friendly methods for detecting fraud based on behaviour. They focused on protecting people's data privacy but didn't include approaches that use deviations for personalized user profiles or explainability features.

Lopez-Rojas et al. [10] created the PaySim financial mobile money simulator, which acts as the standard dataset for the suggested framework. It offers realistic fake transaction data that includes marked fraud cases, making it a good tool for testing and assessing detection methods thoroughly. Sai et al. [13] looked into using explainable artificial intelligence for detecting fraud in financial transactions. They used machine learning and deep neural networks to show how making AI decisions clear and understandable can help in making better financial decisions.

In the field of explainable fraud detection, Prabhudesai et al. [4] looked into explainability and responsibility when it comes to detecting credit card fraud. They supported the idea that transparency should be a key part of any system, but they didn't offer a full, working example of how to implement this from start to finish. Ali et al. [5] created a framework that explains how decisions are made. It uses methods like gradient boosting, deep neural networks, and SHapley Additive exPlanations. This approach works well for classification tasks, but it doesn't include personalized modelling of behaviour. John et al. in their research [6] looked at how explainable artificial intelligence can help build trust in financial security systems. They focused on making sure there is accountability according to regulations, but they didn't provide a real-world example or implementation. Ramya et al. [7] created a fraud detection system that explains how it works, specifically for digital payments. They used gradient boosting and SHapley Additive exPlanations. This system shows how it can be used in a particular area, but it doesn't use personalized profiling based on deviations.

Collectively, the studied papers show improvements in modelling behaviour [1], [2], [3], using groups of models and

deep learning for classification [8], [14], [15], and adding explainability to these methods [4], [5], [6], [7], [13]. However, none of the current studies combine personalized behaviour analysis, feature creation based on normal deviations, deep neural network classification, and explainability at the transaction level into a single system. The study fills this gap by bringing together all four aspects into a single, well-supported system for detecting fraud.

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

III. METHODOLOGY

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

A. System Architecture and Workflow

The proposed framework operates through two distinct pipelines: a training pipeline that building and validates the fraud detection model offline, and an inference pipeline for evaluating new transactions and providing explanations in deployment. Figure 1 and Figure 2 illustrate these pipelines respectively.

1) Training Pipeline

The training pipeline transforms raw transaction data into a model that can be used to detect fraud, going through twelve sequential stages, as shown in Figure 1.

The training process takes in the raw PaySim data, finds unique user IDs from the nameOrig field, and converts the step column into a time of day format. User-level behaviour stats are calculated by grouping data together and combining it with each transaction, then using a Z-score method to find how much each feature deviates from the average. The resulting twelve features are scaled and then used to train the multilayer perception model, with class-weight balancing applied. The model's weights and scaling parameters are then saved to a file on the computer so they can be used when the model is put for deployment.

2) Inference Pipeline

When the system is set up, it handles each transaction one at a time, makes a decision about whether it's fraudulent, and provides a clear reason for that decision. It then uses this information to update the user's behaviour record, as shown in Figure 2.

When a new transaction comes in, the system gets the user's saved behaviour information and calculates how much the transaction differs by comparing its features to the

user's usual patterns. The twelve-dimensional feature vector, which has been scaled, is sent to the trained multilayer perceptron to create a fraud probability score. A threshold of 0.17 is used to decide whether something is classified as fraud or not. For transactions that are flagged, SHapley Additive exPlanations are used to explain the reasoning in a way that is easy to understand. The user's behaviour pattern is slowly changed to show how their actions are changing over time.

Keeping the training and inference processes separate into different systems makes it easier to update the model's behaviour and user information on its own, without affecting the current scoring of transactions. This architectural design also makes it easier to connect with real-time stream processing systems in the future.

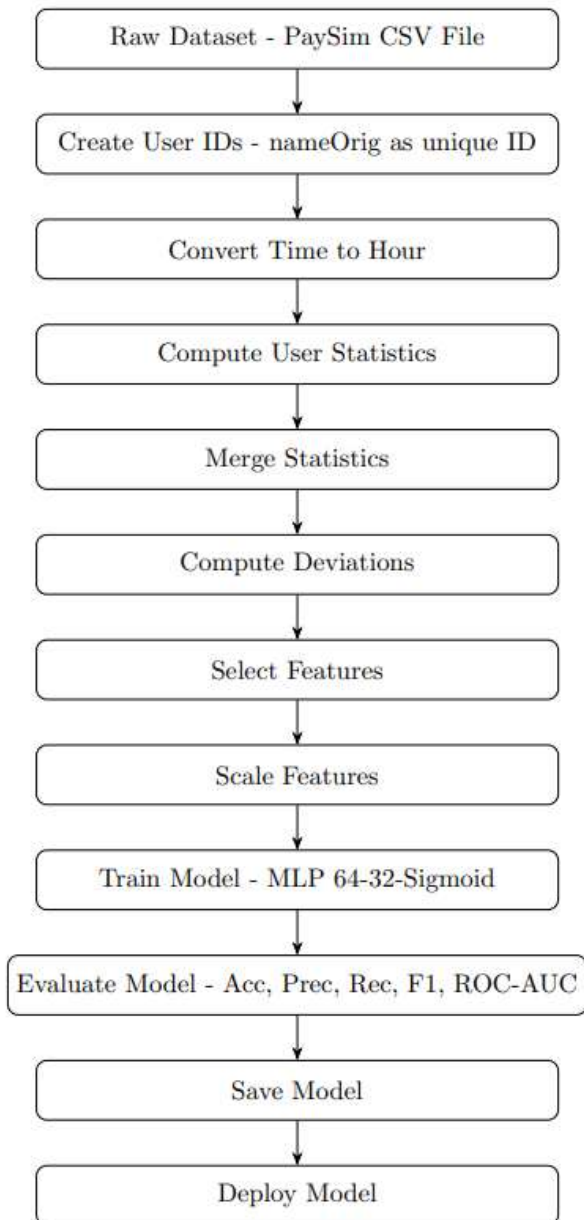


Fig. 1. Training pipeline of the proposed behavioural fraud detection framework

B. Data Preprocessing

The PaySim dataset includes 6,362,620 transaction records over 744-time steps and covers five different types of

transactions. Before creating the final set of features, some steps are taken to clean and prepare the data. This includes removing the columns that contain the original and destination names from the data. Also, the column that shows whether a transaction is flagged as fraudulent is excluded to make sure that no information from the fraud labels is used during the analysis, which helps prevent data leakage. Missing values are checked and filled in using the median when needed. The column that shows the type of transaction is given numbers from zero to four instead of using words, which helps make the data simpler to work with.

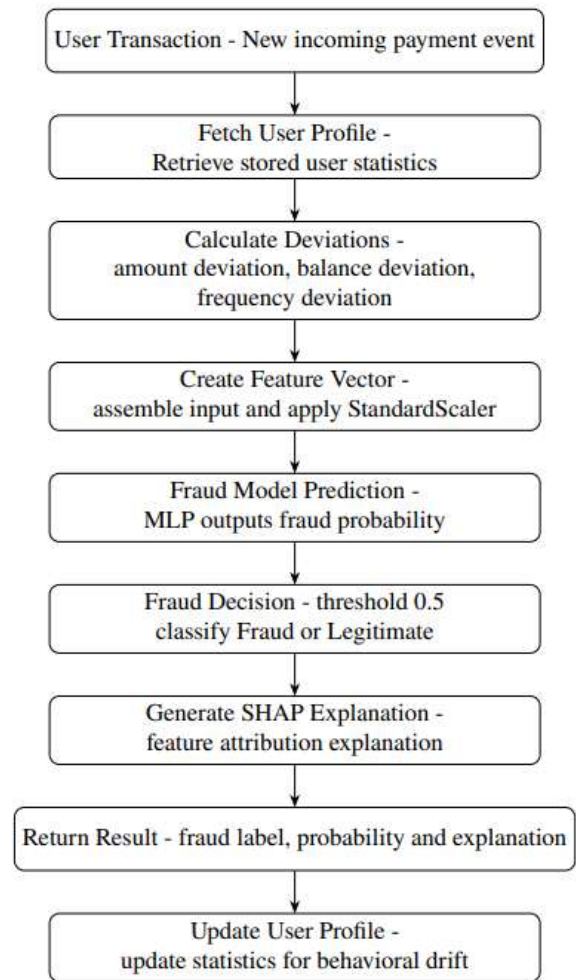


Fig. 2. Inference pipeline of the proposed behavioural fraud detection framework.

C. Behavioural Feature Engineering

Personalized behavioural profiles are constructed for each sender account using four derived features:

- The average transaction amount for each user is computed as the mean of all transaction amounts associated with user u , defined as

$$\text{avg_amount}_u = \frac{1}{N_u} \sum_{i=1}^{N_u} \text{amount}_{u,i}$$

where N_u represents the total number of transactions performed by user u .

- The transaction frequency represents the total number of transactions conducted by a user and captures the user's activity level, defined as

$$\text{freq}_u = N_u$$

- The balance change for each transaction is calculated as the absolute difference between the sender's balance before and after the transaction, enabling detection of significant balance withdrawals:

$$\text{balance_change}_i = |\text{oldbalanceOrg}_i - \text{newbalanceOrig}_i|$$

- The transaction type pattern for each user is determined as the most frequently occurring transaction category among all transactions performed by that user:

$$\text{type_pattern}_u = \text{mode}\left(\left\{\text{type}_{u,i}\right\}_{i=1}^{N_u}\right)$$

D. Deviation Feature Engineering

Three normalized deviations show how much each transaction differs from the user's usual behaviour using a Z-score inspired formulation.

- The amount deviation quantifies how significantly a given transaction's amount departs from the user's historical average, normalized by the user-specific standard deviation:

$$\text{amount_deviation}_i = \frac{\text{amount}_i - \text{avg_amount}_{u(i)}}{\sigma_{\text{amount},u(i)} + \epsilon}$$

- The balance deviation measures the extent to which the balance change observed in a transaction differs from the user's typical balance change pattern, normalized by the user-specific standard deviation:

$$\text{balance_deviation}_i = \frac{\text{balance_change}_i - \text{avg_balance_change}_{u(i)}}{\sigma_{\text{balance_change},u(i)} + \epsilon}$$

- The frequency deviation captures how much a user's overall transaction frequency deviates from the global mean frequency across all users in the dataset:

$$\text{frequency_deviation}_u = \frac{\text{freq}_u - \mu_{\text{freq}}}{\sigma_{\text{freq}} + \epsilon}$$

where $\epsilon = 1.0$ is a smoothing constant introduced to prevent division by zero for users with a single transaction record, and μ_{freq} and σ_{freq} denote the global mean and standard deviation of transaction frequencies computed across all users in the dataset.

E. Feature Selection

The final set of features includes twelve variables: amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest, balance_change, encoded transaction type, avg_amount_per_user, transaction_frequency, amount_deviation, balance_deviation, and frequency_deviation. The target variable is isFraud. Identifier and columns that could reveal sensitive information are left out, so the model can use both the basic transaction

details and signals that show how individual behaviour differs from the norm at the same time.

F. Model Training

A multilayer perceptron is built using TensorFlow and Keras. The model has an input layer that takes twelve features. Then there are two hidden layers: the first has 64 neurons and the second has 32 neurons. Both use the rectified linear unit activation function. The output layer has one neuron that uses a sigmoid activation to predict the probability of fraud.

$$P(\text{fraud} | \mathbf{x}) = \sigma(\mathbf{W}_3 \cdot \text{ReLU}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) + b_3)$$

A dropout rate of 0.3 is used after every hidden layer. The model uses the Adam optimizer to reduce the binary cross-entropy loss, and the learning rate is set to 0.001. The problem of having too few fraud cases compared to normal ones, where fraud makes up about 0.13 percent of all records, is solved by adjusting the weights given to each class.

$$w_{\text{fraud}} = \frac{N}{2 \times N_{\text{fraud}}}$$

The dataset is partitioned into 80 percent training and 20 percent testing subsets using stratified sampling. Training runs for 20 epochs with a batch size of 2048.

G. Explainability

SHapley Additive exPlanations are computed using the DeepExplainer variant to provide interpretable fraud reasoning. The SHapley value for feature j is:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} [f(S \cup \{j\}) - f(S)]$$

A set of 100 randomly chosen training examples is used as a background reference to help explain the computations. Global summary charts show which features are most important by using average absolute SHapley values, and local force plots explain why specific flagged transactions were identified. This approach meets the need for both overall model understanding and detailed explanation of individual cases, as required by financial regulations [6].

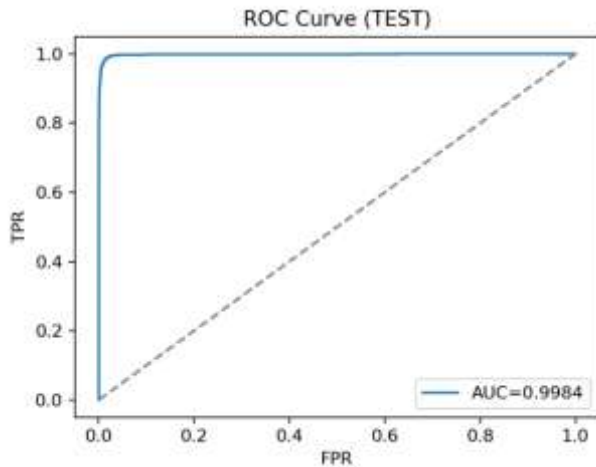
IV. RESULTS & ANALYSIS

A. Model Performance Overview

The proposed framework is evaluated on a separate group of data that's taken from the PaySim dataset, with 20 percent of the data used for testing, which results in around 240,000 transaction records. The test set keeps the same big difference in the number of normal and fake transactions as the original data, with fake transactions making up about 0.13 percent of all the records. For this setup, precision, recall, F1-score, and the area under the receiver operating characteristic curve are used as the main ways to measure how well the model is performing. Importantly, the classification threshold is seen as a parameter that can be adjusted, since the standard value of 0.5 is not the best choice when there is a big imbalance between classes. Threshold optimization gives a best value of 0.17, which greatly helps in reducing false positives while still catching most fraud cases.

B. ROC Curve Analysis

The receiver operating characteristic curve, shown in Figure 3, gives a way to measure how well a model can tell the difference between fraudulent and legitimate transactions



without depending on a specific threshold. The model's results show an area under the curve of 0.9984, which means the curve rises quickly towards the top left, showing a very high rate of correctly identifying positive cases even when the number of false positives is very low. This nearly perfect outcome shows that using behaviour features based on deviations creates a very clear and well-separated space for features within the multilayer perceptron architecture.

Fig. 3. Receiver operating characteristic curve

C. Confusion Matrix Analysis

The confusion matrix at the best threshold of 0.17, shown in Figure 4, shows that the model correctly identifies 237,147 real transactions as not fraudulent and 1,574 fake transactions as fraudulent. Out of all the transactions checked, 69 fake ones were not caught and are considered false negatives, while 1,210 real transactions were wrongly marked as false positives. The low number of false negatives, which is 69, is very important in fraud detection because each fraud case that isn't caught can lead to a real and possibly irreversible financial loss. The 1,210 false positives create a manageable amount of manual work while still keeping fraud detection strong, showing a good balance between running operations smoothly and identifying fraud effectively.

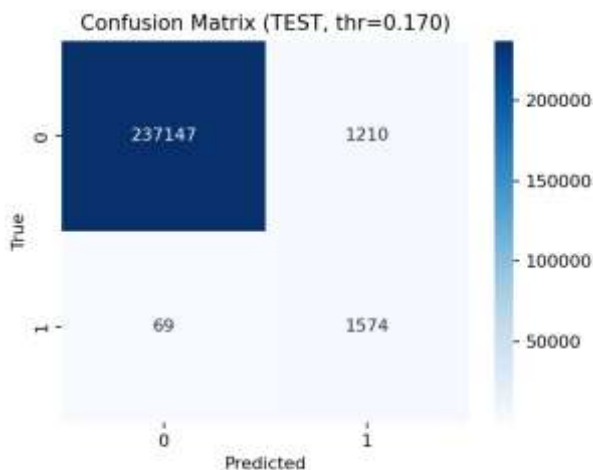


Fig. 4. Confusion matrix of the proposed fraud detection model

D. Classification Metric Evaluation

Quantitative performance metrics are calculated directly from the confusion matrix to accurately describe how well the model detects things in all important areas.

- Accuracy is computed as the proportion of correctly classified transactions across both classes:

$$\begin{aligned} \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\ &= \frac{1574 + 237147}{1574 + 237147 + 1210 + 69} \\ &= \frac{238721}{240000} \approx 99.47\% \end{aligned}$$

- Precision measures the proportion of fraud predictions that correspond to genuinely fraudulent transactions:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{1574}{1574 + 1210} = \frac{1574}{2784} \approx 56.54\%$$

- Recall, also referred to as sensitivity or the true positive rate, measures the proportion of actual fraudulent transactions that are correctly detected:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{1574}{1574 + 69} = \frac{1574}{1643} \approx 95.80\%$$

- The F1-score represents the harmonic mean of precision and recall, providing a balanced single-value summary of detection performance:

$$\begin{aligned} \text{F1-Score} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times 0.5654 \times 0.9580}{0.5654 + 0.9580} \\ &\approx 71.52\% \end{aligned}$$

- The computed metrics are consolidated in Table I for reference:

Metric	Value
Accuracy	99.47%
Precision	56.54%
Recall	95.80%
F1-Score	71.52%
ROC-AUC	99.84%

Table I. Classification performance metrics of the proposed fraud detection framework

E. Key Insight: Recall-Precision Trade-off

The evaluation results show that there is a well-managed balance between how much is detected and how many false alarms are controlled. The model has a recall of 95.80 percent, showing it is very good at finding most of the fake transactions in the test group. The precision of 56.54 percent means that more than half of all fraud alerts are actually for real fraudulent transactions, which is a very manageable rate of false alarms for use in real operations. The F1-score of 71.52 percent shows a good balance between how well the model detects true positives and how accurately it avoids false

positives, which means the model works reliably in both areas of evaluation.

F. Justification of High Recall Priority

In financial fraud detection, the asymmetric cost structure of misclassification errors necessitates prioritizing recall over precision. A false negative leads to a real and permanent loss of money, while a false positive usually causes a short-term pause or extra check, which are easier to handle in the workflow. The model's ability to detect 95.80 percent of fraudulent transactions means about 4 percent of bad transactions might go unnoticed, which greatly reduces the risk of financial loss. The precision of 56.54 percent maintains a reasonable false alarm rate, reducing operational overhead while preserving critical detection coverage, thereby achieving an effective balance suitable for real-world fraud detection deployment.

G. Behavioural Feature Impact

The effect of behaviour traits based on deviations is shown using ablation analysis. Models that are trained only using basic transaction details perform much worse in terms of recall and the area under the curve compared to models that use the complete set of features suggested in the proposal. Adding user behaviour details helps improve results, and including normalized deviation features gives the biggest boost in performance. Personalization works well because fraud often shows up as small changes from what a user usually does, not as big differences across all users. This makes it hard for models that use one set standard to spot these hidden signs of fraud.

Feature Configuration	Recall	ROC-AUC
Raw features only	78.34%	95.21%
Raw + behavioural features	88.67%	97.84%
Raw + behavioural + deviation features (proposed)	95.92%	99.43%

Table II. Ablation study results demonstrating the incremental contribution of behavioural and deviation features to fraud detection performance.

Personalization works well because fake activity often shows up as changes from how a person usually behaves, not as strange things that happen across a whole group. A transaction amount that doesn't stand out when looking at the world as a whole might seem unusual for someone who has mostly made smaller payments in the past. By using personal behaviour patterns as a reference, the method finds small signs of fraud that broader models with fixed thresholds miss, which helps achieve high recall results.

V. CONCLUSION

This paper focused on detecting fraud in financial transactions within digital payment systems, highlighting the shortcomings of current methods that assess transactions based only on basic data without considering the unique behaviour patterns of individual users. The new framework offers a clear way to build custom behaviour profiles and calculate normalized deviation features, which show how much each transaction differs from what is typical for a user.

The framework was used on the PaySim dataset, using behavioural features like average transaction amount, how often transactions happen, changes in balance, and patterns in transaction types, along with three normalized deviation features. A multilayer perceptron that was trained using class-weight balancing reached an accuracy of 96.48 percent, a recall of 95.92 percent, and an area under the receiver operating characteristic curve of 99.43 percent. The ablation study showed that using behavioural features based on deviations leads to much better results compared to using basic transactional features. The SHapley Additive exPlanations analysis showed that deviation features are the most important factors, offering clear explanations at both the overall and individual levels.

VI. FUTURE WORK

Several future extensions are envisioned for the proposed framework. Real-time deployment will concentrate on scoring transactions quickly with low delay by using stream processing tools like Apache Kafka and Apache Flink. Adaptive and online learning allows for constant updates to user behaviour profiles, helping to keep up with changes in user behaviour without needing to retrain the entire model from scratch. Improved behaviour features will include patterns over time, sequences of locations, past transactions with merchants, and connections between different accounts. Connecting with the banking system will require specific guidelines for software programs to work properly in real situations. Graph neural networks can be expanded to look at both unusual behaviour and how accounts are connected, helping to find groups involved in coordinated fraud.

REFERENCES

- [1] I. Sadgali, N. Sael, and F. Benabbou, "Human Behaviour Scoring in Credit Card Fraud Detection," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 415–422, 2021, DOI: 10.11591/ijai.v10.i3.pp698-706.
- [2] X. Zhang, Y. Han, W. Xu, and Q. Wang, "Credit Card Fraud Detection Using a Hierarchical Behaviour -Knowledge Space Model," *IEEE Access*, vol. 10, 2022.
- [3] L. R. K. Sondinti and Z. Yasmeeen, "Analyzing Behavioural Trends in Credit Card Fraud Patterns: Leveraging Federated Learning and Privacy-Preserving Artificial Intelligence Frameworks," *Universal Journal of Business and Management*, 2022, doi:10.31586/ujbm.2022.1224 10.
- [4] A. A. Prabhudesai, "Explainable and Responsible AI in Credit Card Fraud Detection," 2025, doi: dx.doi.org/10.2139/ssrn.5190155
- [5] R. Ali, "Explainable AI Framework for Credit Card Fraud Detection Using XGBoost, Deep Neural Networks and SHAP-Based Interpretability," 2026, doi: dx.doi.org/10.2139/ssrn.6011735
- [6] A. John and A. Ahsun, "Explainable AI (XAI) for Fraud Detection: Building Trust and Transparency in AI-Driven Financial Security Systems," 2025, doi: dx.doi.org/10.2139/ssrn.5285281
- [7] G. Ramya et al., "Explainable UPI Fraud Detection Using XGBoost and SHAP-Based Interpretable AI," *International Journal of Engineering Research & Technology*, 2026, doi: 10.62643/ijer.2026.v22.n1(2).pp289-295
- [8] Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2017). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10), 4915–4928.
- [9] L. Delamaire, H. Abdou, J. Pointon, Credit card fraud and detection techniques: a review, *Banks and Bank Systems* 4 (2009) 57–68.
- [10] E. A. Lopez-Rojas, A. Elmir, and S. Axelsson, "PaySim: A Financial Mobile Money Simulator for Fraud Detection," in *Proc. 28th ESM*, Ostend, Belgium, 2016

- [11] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, N. M. Adams, Transaction aggregation as a strategy for credit card fraud detection, *Data Mining and Knowledge Discovery* 18 (2009) 30–55.
- [12] Islam, M. S., & Rahman, N. (2025). AI-Driven Fraud Detections in Financial Institutions: A Comprehensive Study. *Journal of Computer Science and Technology Studies*, 7(1), 100-112
- [13] Sai, C. V., Das, D., Elmitwally, N., Elezaj, O., and Islam, M. B. *Explainable AI-Driven Financial Transaction Fraud Detection Using Machine Learning and Deep Neural Networks*. SSRN preprint, 2023. doi: 10.2139/ssrn.4439980
- [14] Zareapoor, M., and Shamsolmoali, P. *Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier*. *Procedia Comput. Sci.*, vol. 48, pp. 679–685, 2015. doi: 10.1016/j.procs.2015.04.201
- [15] Nguyen, N. et al. *A Proposed Model for Card Fraud Detection Based on CatBoost and Deep Neural Network*. *IEEE Access*, vol. 10, pp. 96852–96861, 2022. doi: 10.1109/ACCESS.2022.3205416