Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

XAI-Guided Features with Neural Learning Architecture for Anti-Money Laundering Detection in Financial Transactions

Ravi N.[1*], Sathiamoorthy S.[2]

Department of Computer and Information Science, Annamalai University, Annamalai Nagar, India.

E-mail: [1*]nravi2000@gmail.com, [2]ks_sathia@yahoo.com

Orcid ID: [1*]0009-0008-8194-7584, [2]0000-0002-0774-0058

Abstract

Anti-Money Laundering (AML) involves concealing the origin of illegal money through financial crime and the necessity for nations and institutions to develop and implement AML measures. Machine learning (ML) have shown efficiency in detecting suspicious transactions; nonetheless, they do not capture complex non-linear behavior from transactions while maintaining transparency and reliability. In addition, the black box nature of deep learning techniques makes them unsuitable for real-time implementation in financial security applications, as they require transparent decision-making. To address this problem, the current research proposes XAIGFL-NN for Robust AML Detection. The model is designed for integrating domain-aware feature engineering, minority class-preserving sampling, and explainable neural learning to detect suspicious transactions in a large-scale financial environment. A neural network that has been fine-tuned through RMSprop is used to learn discriminatory transaction representations, and SHAP-based interpretability is applied to discover significant transaction features to improve the learning of feature significance. Different from conventional XAI models that utilize explainability for interpretability purposes only, the proposed approach leverages SHAP-based feature engineering to improve both model interpretability and anomaly detection capability. The approach is tested on two popular benchmark datasets, namely IBM AMLSim and PaySim, both of which have heavily skewed class distributions. Extensive comparative outcomes report the encouraging performance of the XAIGFL-NN algorithm over recent methodologies. Therefore, the proposed model is found to be an accurate, interpretable, and robust solution for next-generation AML and financial transaction intelligence systems.

Keywords: Anti-Money Laundering, Class Imbalance, Fraud Detection, Explainable Artificial Intelligence, Feature Learning, Baseline Model.

1. Introduction

Money laundering is the method of transforming illegal money into legitimate money by utilizing various techniques to conceal the money's source. AML is a number of activities,

*** Corresponding Author**

DOI: https://doi.org/10.36548/jtcsst.2026.3.017
Received: 21.05.2026, received in revised form: 24.06.2026, accepted: 16.07.2026, published: 29.07.2026

laws, rules, and procedures developed for detecting and preventing performances that produce income from illicit actions [1]. Black money must be identified by the primary dual steps of placement and layering. Then, it can be complex to inspect its source in the final phase. The requirement for automated software the detection of in detecting suspicious transactions is essential [2]. All financial organizations are accountable for using global regulations. The AML regulations require all financial organizations to monitor, investigate, and report all suspicious transactions [3]. This is required for assuring that the receiver and sender are not registered on blacklists, as it is illegal to do business with them. It is a crucial obligation to prevent money laundering, as a secure bank is at risk and could pay major penalties. Therefore, a practical and intellectual system is required for protecting the banks and monitoring money transactions [4]. With the constant upsurge in transactions, modifying fraud patterns and monitoring landscape, a strong end-to-end architecture utilizing AI that can be needed and supported for precisely identifying suspicious money laundering transactions by decreasing false positives and producing a human interpretation for decision-making by employing Deep Learning (DL) and Machine Learning (ML) techniques [5].

The incorporation of ML and DL techniques has emerged as a major influence in AML approaches, transforming how the financial sectors identifies, prevents, and reports suspicious transactions [6]. However, this technological development also presents particular issues in implementation, computational efficiency, and operational incorporation with present directive measures [7]. The importance of DL and ML in the AML process cannot be underestimated. Finance companies process many transactions every day, producing large amounts of unstructured and structured data, for which conventional rule-based models struggle to ensure efficient analysis. Current money laundering systems are becoming progressively intricate by using advanced technological networks of transactions, the cryptocurrency field, and shell companies to obscure illegal money transactions. This development demands more emerging detection techniques that can proficiently recognize refined patterns and anomalies that may indicate money laundering actions. The execution of ML and DL techniques in AML confirmation signifies a shift from the standard method. The above-mentioned innovative systems are used for analyzing previous transaction information, customer behavior patterns, and external information sources concurrently, allowing more precise identification of suspicious activities and lessening false positives. Financial organizations are increasingly using unsupervised and supervised learning methods to effectively improve regulatory reporting procedures, customer risk valuation, and transaction monitoring.

This paper proposes XAI Guided Feature Learning with a Neural Network (XAIGFL-NN) for a robust Anti-Money Laundering identification network. The proposed method is an explainable and technically robust fraud detection framework that integrates deep learning with SHAP-based interpretability to enhance classification performance. The architecture incorporates SHAP-based comprehensibility, minority-class–preserving undersampling, a baseline classifier, and domain-based feature extraction to highlight data imbalance and optimize identification performance. Experimental outcomes on benchmark datasets validate improved robustness and accuracy in comparison with current models. The key contributions of this proposal are presented here:

- A minority-class–preserving undersampling tactic is employed to manage severe class imbalance while maintaining all fraudulent instances. Domain-based and environment-aware feature extraction is then accomplished to efficiently acquire anomalous transaction patterns.

- A fully connected neural network (FCNN) enhanced using the RMSprop algorithm is used as the baseline classifier. This optimization improves convergence efficiency and learning stability from highly imbalanced data settings.
- SHAP is utilized to offer clear explanations of the method's estimation by detecting key features influencing fraud identification. By incorporating SHAP with FCNN, the proposed architecture achieves superior transparency and dependable classification performance.
- The presented XAIGFL-NN architecture is assessed on the IBM AMLSim and PaySim benchmark datasets from Kaggle. Widespread comparative outcomes validate improved detection performance and comprehensibility compared to current AML identification methods.

Despite the lack of an algorithm that learns from scratch in the introduced XAIGFL-NN system, the novelty of this work resides in introducing a well-designed framework for the AML detection system that integrates the application of complementary techniques. The existing literature on AML, tends to study either one aspect of AML such as class imbalance, DL-based classification, or interpretability, individually. Contrary to existing approaches, the proposed architecture integrates the following techniques: minority class preserving undersampling, domain and context-based feature engineering, FCNN optimized by the RMSprop optimizer, and SHAP-based interpretability in a single framework. Such integration enables the model to effectively highlight the severity of the class imbalance problem, extract behavior patterns of each transaction, provide constant training of the neural network, and explain predictions. Additionally, the proposed framework is thoroughly evaluated using two benchmark datasets – IBM AMLSim and PaySim – proving the robustness of the approach across diverse transaction environments.

2. Literature Review

The following is a discussion of relevant literature regarding AML detection, where the primary knowledge gaps in this domain have been identified. Olaniyi et al. [8] introduced a novel approach to the GNN-XAI framework, which is modified for multi-level AML detection with accuracy, scalability, compliance with regulations, and privacy protection in mind. It is a heterogeneous Graph Attention Network based approach for entity relationships, transaction flow, temporal interactions, and device connectivity in complex financial systems. LIME, SHAP, and attention-based explanations enable auditable, understandable, supervisable, and compliant solutions. Lin et al. [9] presented ChronoWave-GNN, a graph neural network architecture developed through the conception framework of time-frequency feature learning. By integrating wavelet-driven spectral disintegration with a time-based encoder process, the author's technique acquires multi-scale and nonstationary patterns intrinsic to illegal financial movement. This two-field perspective improves the descriptive measurements of graph-based representations. Moreover, training techniques like smoothing of labels and cosine learning-rate scheduling have been integrated to improve steadiness from extreme class imbalance. Sidiq and Wondaferew [10] introduce the essential role played by AML operations in the worldwide financial industry. It highlights the disadvantages of traditional AML networks that expose higher rates of false positives (FP) and exhibit limited sophistication to expose complicated money laundering schemes. To avoid these kinds of issues, the author introduces an innovative AML method, which leverages link analysis utilizing DL models. These structures improve the network's ability to detect unusual behavior through investigating the effect and relationship among financial transaction networks.

Wang et al. [11] present a Reinforcement Learning (RL) improved, multi-relation, attention graph-aware architecture to identify AML and illicit transaction practices. On one side, a graph-aware, data-based state implemented the long-lasting dependencies and associations among transaction graph vertices. Resemblance between graph nodes separates the topological graph into three sub-graphs. The integration of converging nodes enhances global, local, and contextual data. Simultaneously, utilizing repeated nodes across the sub-graphs improves interaction among them, decreases intra-class ambiguity, and emphasizes inter-class variations. Wu et al. [12] explore the increasing difficulty of transaction activities and the highly hidden nature of money laundering methods in present financial AML environments. They present a Transformer-driven risk surveillance method for AML. The technique is based on the transaction sequence model and incorporates operational data. An environment-aware classifier could be presented to allow precise detection and risk ranking of higher complexity. The method initially uses feature engineering and Positional Encoding (PE) for every transaction. Afterward, it employs the numerous Transformer states to acquire distant behavior correlations. Yu et al. [13] introduce the usage of unsupervised learning techniques in cross-border AML networks, with attention to rule enhancement contrastive learning models. Five DL techniques, ranging from basic CNNs to a hybrid CNN- GRU framework, have been developed and tested to measure their performance in identifying suspicious transactions. The self-reliant hybrid Convolutional-Recurrent Neural Integration Model (CRNIM) approach, in particular displayed improved effectiveness.

Wan and Li [14] presented a money laundering and fraud identification method that depends on DL, recommended as MDGC-LSTM. The technique integrates the usage of a dynamic graph convolutional network (MDGC) and a long short-term memory (LSTM) model to effectively detect AML in economic transactions. The approach makes dynamic graph snapshots with symmetric spatio-temporal algorithms according to transaction data, indicating transaction nodes and flow of money as graph edges and nodes, correspondingly, and efficiently acquires the relation among temporary and spatial frameworks, therefore reaching the dynamic estimation of fraudulent transactions. Karangara and Subbagari [15] proposed a comprehensive investigation of the usage of optimization-based ML models for the filtration and identification of money laundering activities, discovering a range of aspects like data mining, AML policies, supervised and unsupervised learning structures, behavioral modeling, link measurement, anomaly identification, risk score, and geographical applications. The author eventually focused their attention on different extremely relevant articles, emphasizing the contribution of DL approaches in attaining alliance with their a motives. Table 1 presents an overview of existing research works, including their aims, techniques, datasets, results, and limitations in AML detection.

Table 1. Overview of related works

Authors	Research Aim	Techniques	Datasets	Results
Olaniyi et al. [8]	To create an explainable deep learning system for detecting financial fraud.	GNN and XAI models	IEEE-CIS Fraud Detection, Kaggle Credit Card Fraud, and Elliptic Bitcoin datasets.	Precision of 89.3%.
Lin et al. [9]	To develop a graph-based DL method for detecting cryptocurrency financial crimes.	Transformer and GNN techniques	Elliptic dataset.	Accuracy of 0.9802.

Sidiq and Wondaferew [10]	To create an AML approach utilizing graph connectivity and centrality for precise money laundering detection.	GNN and GCN methods	AMLSim dataset.	Achieves better results.
Wang et al. [11]	To develop an enhanced graph-aware model for robust AML and illegal transaction detection.	GCM and DQN approaches	Elliptic and RD Elliptic datasets.	F1-score of 93.85% and 94.39%.
Wu et al. [12]	To develop a model that integrates transaction sequences and graph structure for accurate AML risk detection.	Transformer	Elliptic dataset.	Accuracy of 87.3%.
Yu et al. [13]	To apply unsupervised models for intelligent, adaptive detection of cross-border money laundering.	CNN and GRU methodologies	Elliptic dataset.	Accuracy of 95.6%.
Wan and Li [14]	To develop the dynamic prediction of money laundering using spatiotemporal transaction graphs.	MDGC and LSTM models	Elliptic dataset.	Macro-F1 score of 0.25%.
Karangara and Subbagari [15]	To explore and enhance AML detection using optimization-driven techniques.	Machine Learning techniques	AML dataset.	Gain better accuracy.

2.1 Research Gaps

Although current developments in AML recognition have addressed numerous research gaps, many still remain. More conventional strategies struggle with extremely imbalanced financial transaction information, causing worse identification of minority-class suspicious activities. Several algorithms depend on black-box neural systems, presenting inadequate interpretability for the financial decision-making process. Additionally, standard XAI methodologies are frequently utilized only for post-hoc explanations instead of enhancing feature refinement and model learning. Existing models face difficulties in efficiently extracting risk transaction patterns in wide-scale financial settings. Moreover, there is inadequate incorporation of explainability with feature learning to jointly enhance both transparency and performance.

3. Proposed Model

The proposed XAIGFL-NN architecture presents an innovative technique for AML detection by incorporating SHAP-guided feature learning with deep neural networks. Figure 1 demonstrates the entire workflow of the proposed XAIGFL-NN method. As shown in Figure 1, the presented architecture addresses severe class imbalance with a minority-class preserving undersampling technique, followed by context-aware and domain-driven feature engineering to improve the representation of transactional characteristics. The engineered features are then preprocessed by categorical-to-numerical encoding, feature scaling, and missing value handling to generate a model-ready dataset. Subsequently, the preprocessed features are utilized to train the FCNN methodology, optimized with RMSprop for AML classification.

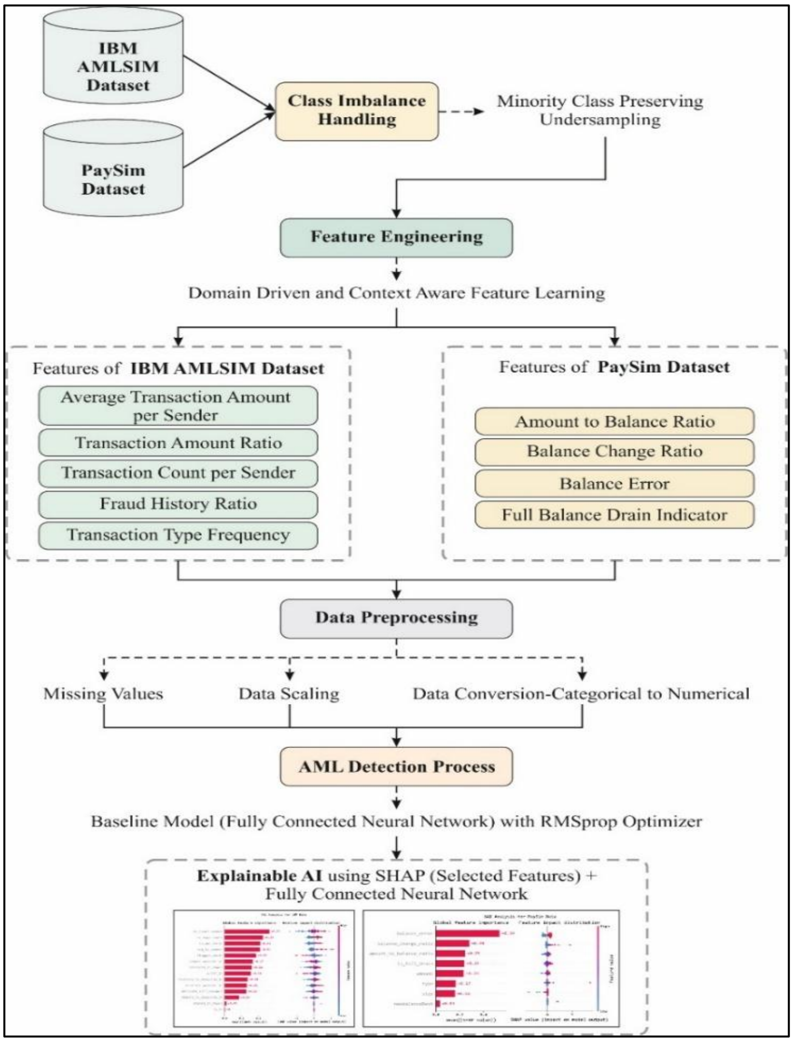

Figure 1. Overall workflow of the proposed XAIGFL-NN framework

After model training, SHAP is applied as a post-hoc explainability method to quantify the contribution of every input feature and interpret the model's predictions. The outcomes of SHAP feature importance rankings provide insights into the influence of individual features on the classification decisions, thus enhancing the transparency and interpretability of the presented architecture. By combining imbalance handling, domain-driven feature engineering, deep learning-based classification, and explainable AI within a unified pipeline, the proposed framework delivers an accurate and interpretable solution for AML detection.

3.1 Data Preparation

In this section, the proposed method is applied to two benchmark datasets, which are first described in detail. The datasets then undergo class imbalance handling, followed by a feature engineering process to extract domain-relevant and context-aware attributes. Finally, data preprocessing is performed to ensure the data is ready for model training and evaluation.

3.1.1 Dataset Details

This study employs two benchmark datasets: the IBM AMLSIM Example Dataset [16] and the PaySim Mobile Money Dataset [17]. A brief description of each dataset is provided below.

IBM AMLSIM Example Dataset: This dataset is an artificial dataset of banking transactions created for anti-money-laundering analysis. It comprises normal and abnormal transactions among accounts, simulating real-life patterns. The data are synthetic but realistic, enabling validation of fraud identification approaches. It is primarily utilized to create and benchmark an AML ML technique. The total features are (TX_ID, SENDER_ACCOUNT_ID, RECEIVER_ACCOUNT_ID, TX_TYPE, TX_AMOUNT, TIMESTAMP, ALERT_ID, SENDER_CUSTOMER_ID, SENDER_INIT_BALANCE, SENDER_COUNTRY, SENDER_ACCOUNT_TYPE, SENDER_IS_FRAUD, SENDER_TX_BEHAVIOR_ID, RECEIVER_CUSTOMER_ID, RECEIVER_INIT_BALANCE, RECEIVER_COUNTRY, RECEIVER_ACCOUNT_TYPE, RECEIVER_IS_FRAUD, RECEIVER_TX_BEHAVIOR_ID, isFraud).

PaySim Mobile Money Dataset: There is a limited availability of public datasets on financial services, particularly in the emerging domain of mobile money transactions. Financial datasets are significant to some analysts and specifically to us executing a study in the fraud identification domain. The component of the issue is the inherently private nature of financial transactions, leading to no publicly available datasets. We propose a synthetic dataset created utilizing the simulator named PaySim as a method to address this concern. PaySim applies aggregated data through the private dataset to create an artificial dataset that mimics the usual operation of transactions and introduces malicious behavior to subsequently assess the proficiency of fraud identification models. The total features are (step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud).

3.1.2 Class Imbalance Handling using Minority-Class–Preserving Undersampling

This method is used for preprocessing and balancing large-scale fraud datasets through controlled undersampling. It preserves the minority class and the balances data by maintaining all minority-class instances while specifically decreasing the number of majority-class instances [18]. In contrast to random undersampling, this technique removes only redundant or minimally useful majority instances, thereby supporting key decision boundaries and class-discriminative patterns.

Considers the dataset is represented by $D_{maj} \cup D_{min}$, where $| D_{maj} | \gg | D_{min} |$.

The balanced dataset can be expressed as,

$$D' = D_{min} \cup D'_{maj}, | D'_{maj} | = \alpha | D_{min} | \quad (1)$$

Where α represents the parameter that controls the undersampling rate and enables dynamic adjustment of the class imbalance ratio. Instances of the majority class have been selected according to the criteria of clustering or distance/density criteria to verify the demonstrative feature space. To increase the size of the minority class and reduce algorithm has been applied computational costs by decreasing class skewness and bias in the majority class.

The undersampling rate has been determined to ensure that enough legitimate transactions remain when reducing the majority class dominance. Rather than randomly dropping majority class instances, representative legitimate transactions have been kept to maintain the distribution of transactionse. Figure 2 illustrates the problem of class imbalance in the dataset.

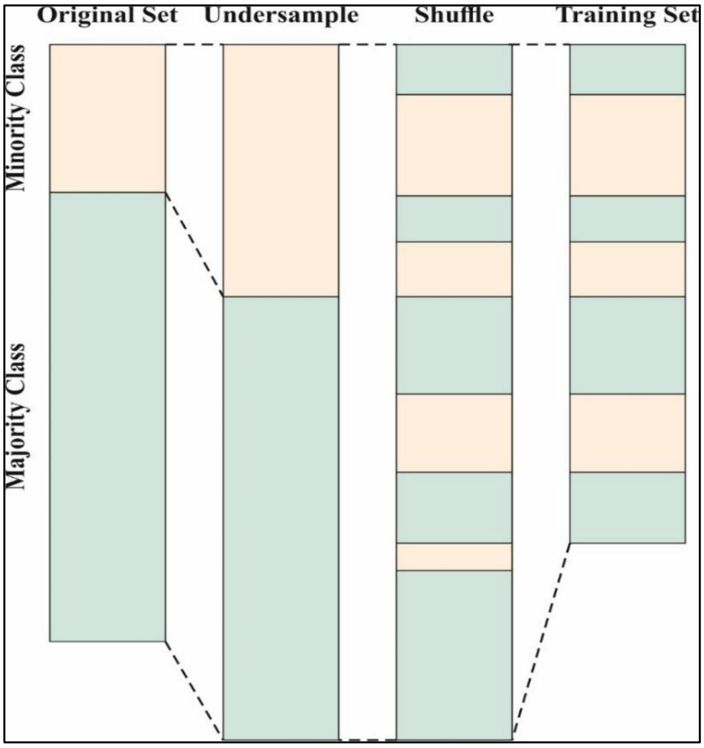

Figure 2. Class imbalance

A majority-to-minority ratio of 2:1 (minority-to-majority 1:2) has been utilized to strike a balance between efficient computation and information retention. The sampling technique retains a significant percentage of legitimate transactions by reducing class dominance, thus enabling the classifier to learn the characteristic attributes of both classes. Hence, the sampling technique eliminates the issue of class bias without sacrificing too much information from the majority class instances.

The IBM AMLSim and PaySim datasets have shown a class imbalance problem where the number of legitimate transactions is significantly higher than the number of money laundering cases. To solve this problem, the Minority class preserving undersampling technique was employed to reduce the dominance of the majority class while retaining all the instances of the minority class. The sampling technique preserves representative patterns of the legitimate transactions and, thus, minimizes information loss while ensuring the integrity of the financial transaction data is retained. In addition, the sampling technique was applied only to the model training phase.

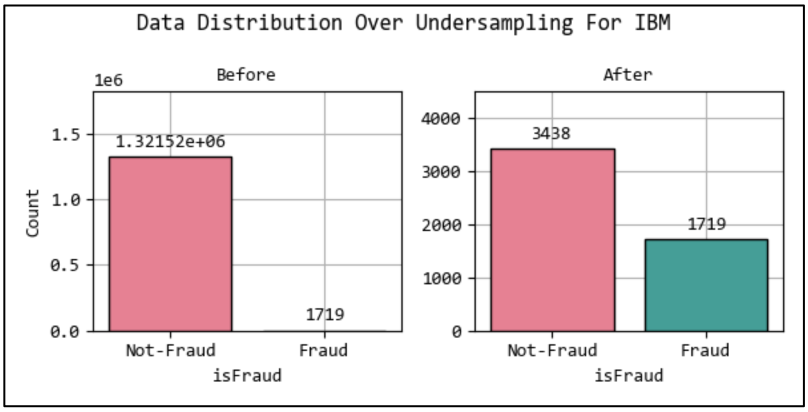

Figure 3. Data distribution over undersampling for IBM data’

Figure 3 shows the class distribution of the IBM transaction data both before and after the undersampling process. Initially, there is a huge imbalance in the dataset because there are nearly 1.32 million Not-Fraud instances while Fraud instances amount to only 1,719. This causes any technique to be inclined towards the common class, making it incapable of identifying fraud patterns. After the undersampling procedure, the Not-Fraud instances are reduced to 3,438, whereas the Fraud instances remain at 1,719, which is a significant improvement in the ratio of class distribution.

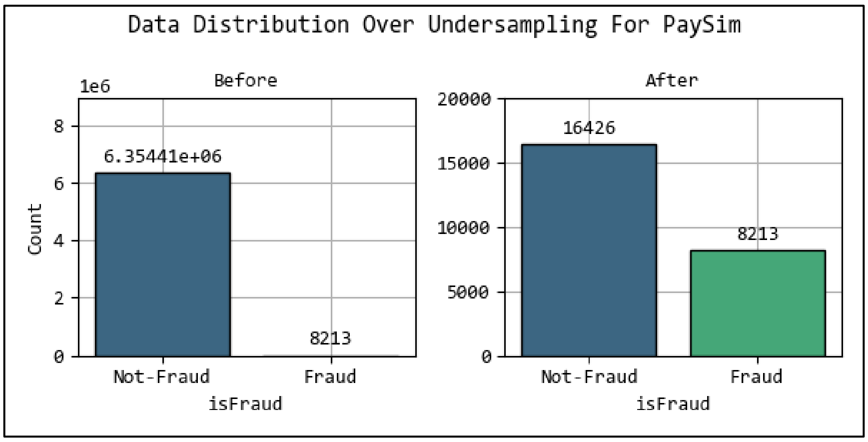

Figure 4. Data distribution over undersampling for PaySim data

Figure 4 illustrates the class imbalance before and after applying undersampling to the PaySim dataset. The dataset exhibits a higher imbalance before undersampling, with approximately 6.35 million non-fraud cases compared to 8,213 fraud cases. This disparity can bias methods towards the majority class. After applying undersampling to the PaySim dataset, the number of non-fraud cases become 16,426, while the number of fraud cases remains at 8,213 and achieving a very high balance in terms of class ratio.

3.1.3 Dataset Construction: Domain-Driven and Context-Aware Feature Learning

A significant aspect of this project is the development of custom behavioral features that identify abnormal transaction patterns, as opposed to relying solely on raw features.

1. **Custom Features for IBM AMLSIM Dataset**

- **Average Transaction Amount per Sender (avg_tx_sender):** Captures the normal spending behavior of each sender.
- **Transaction Amount Ratio (tx_amt_ratio):** Calculates the transaction size relative to the sender's balance, highlighting abnormal transactions.
- **Transaction Count per Sender (tx_count_sender):** Identifies unusually high transaction. frequencies.
- **Transaction Type Frequency (tx_type_count):** Detects over-usage of specific transaction categories, which are often related to fraud. These features are encoded by sender behavior, financial contexts, prior risks, all of which can be crucial for AML detection.

2. Custom Features for PaySim Dataset

- **Amount-to-Balance Ratio (amount_to_balance_ratio):** This is employed for identifying transactions that constitute a significant portion of account balances.

- **Balance Change Ratio (balance_change_ratio):** This can be measured to assess the reliability of the balance change and transaction amount.

- **Balance Error (balance_error):** This detects variations between predicted and actual balances.

- **Full Balance Drain Indicator (is_full_drain):** This is utilized for flagging transactions that drain ≥90% of the account balance. Such features highlight account-draining behavior and transaction reliability, which are robust indicators of fraud in mobile money systems.

3.1.4 Data Preprocessing

Data pre-processing comprises missing-value handling, feature scaling, and categorical-to-numerical conversion to ensure model readiness. preprocessing is accomplished to make the data more crucial, and to ensure the model learnS proficiently [19]. This can be completed in numerous steps:

- **Encoding Categorical Variables:** The transformation of categorical features into numerical representations is achieved by utilizing one-hot encoding. this method converts all category values into new categorical columns and allocates a new binary value (1/0) to indicate the binary values.

- **Data Scaling:** Data normalization is executed for standardizing the data and converting it to have unit variance and a zero mean. This phase is crucial to eliminate the impact of various feature scales, thus simplifying the training process for models. The advantages of normalization comprise rapid convergence during model training, as it eliminates variances in feature magnitudes, thereby allowing gradient descent techniques to more rapidly determine the global optimum. Moreover, it expands model performance by diminishing the effect of differing feature scales, which can be mainly useful for gradient-based optimization techniques. Standardization also supports avoiding numerical variability, specifically in datasets with significant feature modifications. Likewise, in the Transformer architecture, both the target and feature variables have been standardized by employing "StandardScaler", with the statistical data of the corresponding variables. The mathematical expression of data normalization is given below,

$$X_{standardized} = \frac{X - mean(X)}{std(X)} \quad (2)$$

  Here, the terms $mean(X)$ and $std(X)$ signify the mean and standard deviation (SD) of the target or feature variable, respectively.

- **Managing Missing Data:** This is handled by the basic imputer-class introduced from Scikit-learn. This can be changed for all columns' means for the whole set of missing values or numbers. This imputation preserves the integrity and size of the dataset.

3.2 Baseline Model for AML Detection

In this section, an FCNN was deployed as the baseline classifier because the input data consist of structured, domain-driven feature vectors derived from financial transactions rather than graph-structured networks or sequential transaction histories. FCNNs are well suited for learning nonlinear relationships from tabular feature representations when maintaining computational efficiency and constant optimization. The RMSprop optimizer additionally enhances convergence behavior and learning stability under imbalanced classification settings. However, the primary motive of this study is to establish an interpretable AML detection architecture through incorporating efficient feature engineering with SHAP-based explainability, allowing transparent decision-making without requiring complex graph or sequence modeling. A FCNN is also called a feedforward neural network (FFNN), is an important DL method that can be employed for learning intricate correlations among target outputs and input features [20]. The term “fully connected (FC)” has been utilized for the fulfillment of CNNs, which can be explained in the next section. In the FCNN architecture, one layer is interconnected with all the nodes in the following layer. The input data is entered into numerous FC layers, where the network learns different weighted integrations of the input features. Every interconnection among nodes is related to a weight, and every individual node has an exclusive bias term, thus allowing it to change its activation self-reliantly. Such weights and biases have been modified during the training process to decrease the classification errors. The last layer outputs are the probability indication. While the FCNN architecture has a simple structure when associated with the CNN framework, it is a robust tool for gaining the non-linear correlations among target output and input data, thereby making it an adaptable selection for the classification process. limitation of an FCNN is that, the model cannot inherently use the sequential types of the input, UNLIKE CNNs that excel in capturing temporal and spatial patterns. The presented architecture uses an FCNN as the classification methodology because of the structured tabular nature of the IBM AMLSIM and PaySim datasets after feature engineering. The engineered features capture transaction-level and behavioral characteristics, enabling the FCNN to effectively learn nonlinear relationships without requiring explicit graph representations.

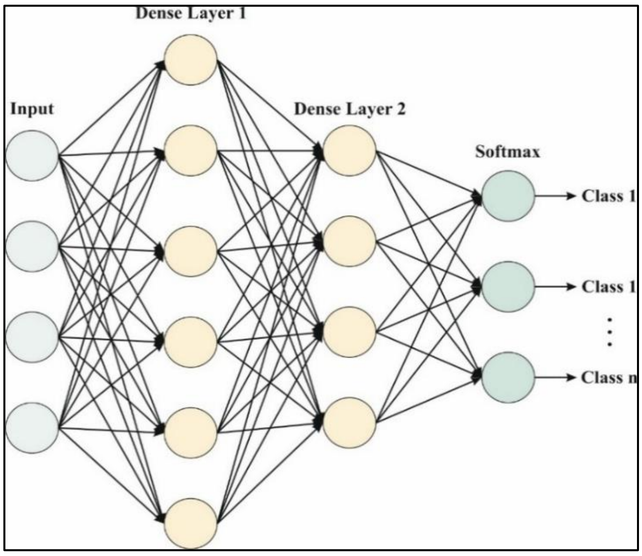

Figure 5. Structure of FCNN model

The FCNN is a simple neural network (NN) framework. Figure 5 illustrates the infrastructure of the FCNN. The major elements of the FCNN are neurons that can be fixed in various layers. An input layer has several neurons, equal to the quantity of inputs to the model (also called features). It is succeeded by variable hidden layers (HL) and an output layer that comprises a number of neurons. The computational expression of the output y_j with a neuron and input data x_i can be represented in Eq. (3),

$$Y_j^{(k)} = f\left(b^{k-1} + \sum_{i=1}^{m_{k-1}} w_{i,j}^{(k)}\, y_i^{(k-1)}\right) \tag{3}$$

Where the term $y_j^{(k)}$ refers to the output with respect to the j^{th} neuron and k^{th} layers, m_{k-1} represents the count of neurons in $the\ k-1$ (previous) layer, $w_{i,j}^{(k)}$ represents the weights related to the i^{th} neuron of the preceding layer, and b^{k-1} denotes the bias connected with the preceding layer. The activation function can be indicated as Rectified Linear Unit $(ReLU\ (z)\ =\ max\ (0, z))$. The function f remains the activation function that presents non-linearity for the output neurons.

The RMSprop (Root Mean Square Propagation) constitutes an adaptive learning rate (ALR) optimization method, which can be developed for increasing Stochastic Gradient Descent (SGD) and overcoming the strong decrease in learning rates of AdaGrad [21]. The major intention of this technique is to accelerate optimization by decreasing the number of operational estimations needed to reach the local minimum. The gradient can be separated by the square root of the mean, which preserves the weight moving average of squared gradients. The mathematical form is Eq. (4), represents a function value as given by $v(w, t)$ with state w with respect to time t.

$$v(w, t) \coloneqq \gamma v(w, t-1) + (1-\gamma)(\nabla Q_i(w))^2 \tag{4}$$

where the parameter γ refers to the discount factor that matches the instant reward with the predicted upcoming rewards, Greater values offer higher-level weights to imminent rewards, and its mathematical form is $v(w, t-1)$. It signifies the value of the system with respect to state w and the preceding time step $(t-1)$. $(1-\gamma)$. Additionally, the weight significance of the instant reward when related to upcoming rewards (w). This is a state-value operation that can allocate the values for every state. ∇Q. The computational representation of specific complexity is indicated as given below,

$$w \coloneqq w - \frac{\eta}{\sqrt{v(w, t)}} \nabla Q_i(w) \tag{5}$$

The engineered features were mathematically formulated to capture transaction behavior and account-level characteristics associated with money laundering activities. For the IBM AMLSIM dataset, the engineered features are defined as:

$$\text{Average Transaction Amount per Sender} = \frac{\sum_{i=1}^{N} Amoun\ t_i}{N} \tag{6}$$

$$\text{Transaction Amount Ratio} = \frac{Transaction\ Amount}{Average\ Transaction\ Amount\ per\ Sender} \tag{7}$$

$$\text{Transaction Count per Sender} = N_{sender} \tag{8}$$

$$\text{Transaction Type Frequency} = \frac{Frequency\ of\ Transaction\ Type}{Total\ Transactions} \tag{10}$$

Similarly, for the PaySim dataset, the engineered features are defined as:

$$\text{Amount-to-Balance Ratio} = \frac{Transaction\ Amount}{Old\ Balance_{Origin}} \tag{11}$$

$$\text{Balance Change Ratio} = \frac{OldBalance_{Origin} - NewBalance_{Origin}}{OldBalance_{Origin}} \tag{12}$$

$$\text{Balance Error} = |\ OldBalance_{Origin} - Transaction\ Amount - NewBalance_{Origin}\ | \tag{13}$$

$$\text{Full Balance Drain Indicator} = \begin{cases} 1, & \text{if } NewBalance_{Origin} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

These engineered features improve the representation of transactional and behavioral characteristics, allowing the proposed FCNN model to learn discriminative patterns related to money laundering activities. Domain-based and context-aware feature extraction comprises Average Transaction Amount per Sender, Transaction Amount Ratio, Transaction Count per Sender, and Transaction Type Frequency. These features are derived exclusively from transaction attributes and behavioral patterns available before model prediction, thereby preventing label-dependent feature construction.

Data Leakage Prevention: To ensure a fair and realistic evaluation, all engineered features were derived from transaction attributes and historical behavioral information available before the prediction of the target transaction. No feature was constructed using the ground-truth fraud labels or post-event information. Moreover, the training and testing datasets were strictly separated throughout feature engineering and model development, thus preventing information leakage among data partitions.

3.3 XAI Enhanced AML Detection

Finally, a hybrid feature selection strategy is adopted, combining Top-K SHAP-ranked features with customized, domain-relevant features. SHAP is an interpretability architecture that depends on cooperative game theory and allocates significant values to every input feature in a specified prediction [22]. After training the FCNN model, SHAP was applied as a post-hoc explainability technique to estimate the contribution of each input feature to the prediction. The mean absolute SHAP values were used to rank feature importance, enabling interpretation of the model's decision-making process. SHAP was employed solely for feature importance analysis and model interpretability and was not used as a feature selection algorithm.

The technique calculates the SHAP values, which confirm a better distribution of the prediction between input features, along with their minimal contributions. For model f and input sample x, the SHAP value ϕ_j is represented to the feature j, and the mathematical form is given by,

$$\phi_j = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(M - |S| - 1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)] \tag{15}$$

Here, the parameter N refers to the whole feature set, M indicates the total counts of features, S signifies the whole feature set without i, and $f_x(S)$ means the model prediction for feature subset S.

SHAP offers global and local interpretability for ML methods. Local interpretability describes distinct predictions by providing model outcomes for every input feature, thus supporting the prediction of individual data points by assigning output for the exact contribution of all features. Global interpretability is accomplished by combining SHAP values to evaluate the total rank and implication of every feature, which gives deeper insights into the major impact of features on the model's prediction. Moreover, the deep explainer evaluates feature imputations that integrate the model's gradients with contextual distributions for estimating Shapley values, thus indicating the minimal contribution of every feature with respect to the baseline input.

In this method, SHAP values quantify the involvement of every feature in fraud prediction, and subsets of Top-K SHAP-chosen features are utilized as input to an FCNN classifier. To maintain explainability and semantic relevance, domain-critical features are identified by employing proficient and contextual data that can be clearly preserved with SHAP-ranked features. For reasonable and constant assessment, a similar NN framework is the baseline model and comprises the optimization algorithm, regularization method, and similar activation functions utilized for classification. The above-mentioned architecture ensures that it offers outcome enhancements and attributes for XAI-guidedFS instead of architectural variations, thus leading to a strong, transparent, and explainable AML detection model.

4. Experimental Result Analysis and Discussion

The performance of the proposed XAIGFL-NN approach is evaluated using two benchmark datasets: the IBM AMLSIM Example Dataset and the PaySim Mobile Money Dataset. These datasets enable a comprehensive assessment of the model’s effectiveness across diverse transaction patterns and realistic financial scenarios. Table 2 presents the parameter settings of the proposed methodology.

Table 2. Parameter setting

Method	Parameter	Value
Class Balancing	Class Ratio	0.043056
Data Scaling	Standardization	StandardScaler
Missing Data	Imputation	Mean
Feature Engineering	Domain Features	5 (IBM), 4 (PaySim)
Neural Network	Hidden Layers	2
Activation	Function	ReLU
Output Layer	Activation	Softmax
Optimizer	Algorithm	RMSprop
Learning Rate	Initial LR	0.001
Batch Size	Batch Size	32
Training	Epochs	50

Figure 6 denotes the SHAP-based XAI analysis for the IBM fraud technique, displaying how features influence predictive results. tx_type_count, tx_count_sender, avg_tx_sender, tx_amt_ratio, and flagged_ratio are the most influential features with the maximum SHAP influence. The SHAP distribution specifies how feature values push estimation to fraud or non-fraud verdicts. Chosen vital features comprise tx_type_count, tx_count_sender, avg_tx_sender, tx_amt_ratio, flagged_proportion, RECEIVER_IS_FRAUD, ALERT_ID, and RECEIVER_TX_BEHAVIOR_ID. From the PaySim dataset, the selected features contain step, type, amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest, isFraud, amount_to_balance_ratio, balance_change_ratio, balance_error, and is_full_drain.

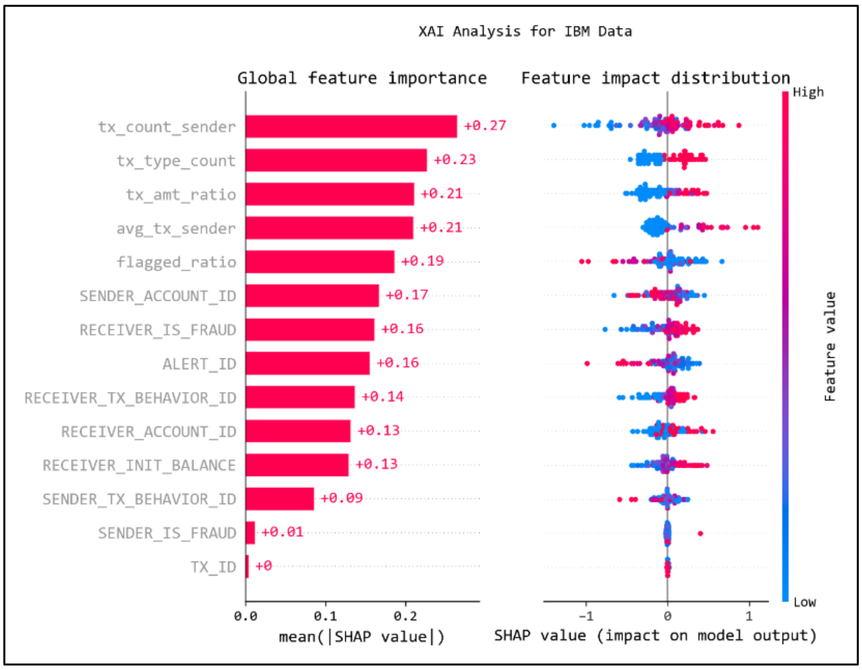

Figure 6. XAI analysis for IBM data

Figure 7 illustrates SHAP-driven XAI analysis for the PaySim fraud method, representing how features impact predictive results. balance_change_ratio, balance_error, amount_to_balance_ratio, is_full_drain, and amount are the most influential features, underlining unusual balance behavior as main fraud signals. The SHAP distribution displays how feature values push estimations to fraud or non-fraud decisions. Chosen vital features comprise balance_change_ratio, balance_error, amount_to_balance_ratio, is_full_drain, amount, step, and type.

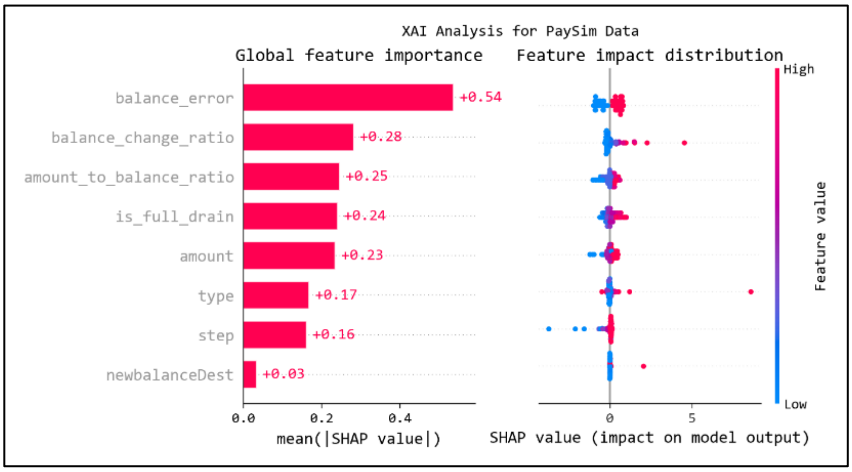

Figure 7. XAI analysis for PaySim data

Figure 8 represents the confusion matrices produced by the XAIGFL-NN model for both baseline and XAI techniques according to the TRAP/TESP split of IBM data. The results indicate that the XAIGFL-NN technique effectively detects and classifies all classes accurately.

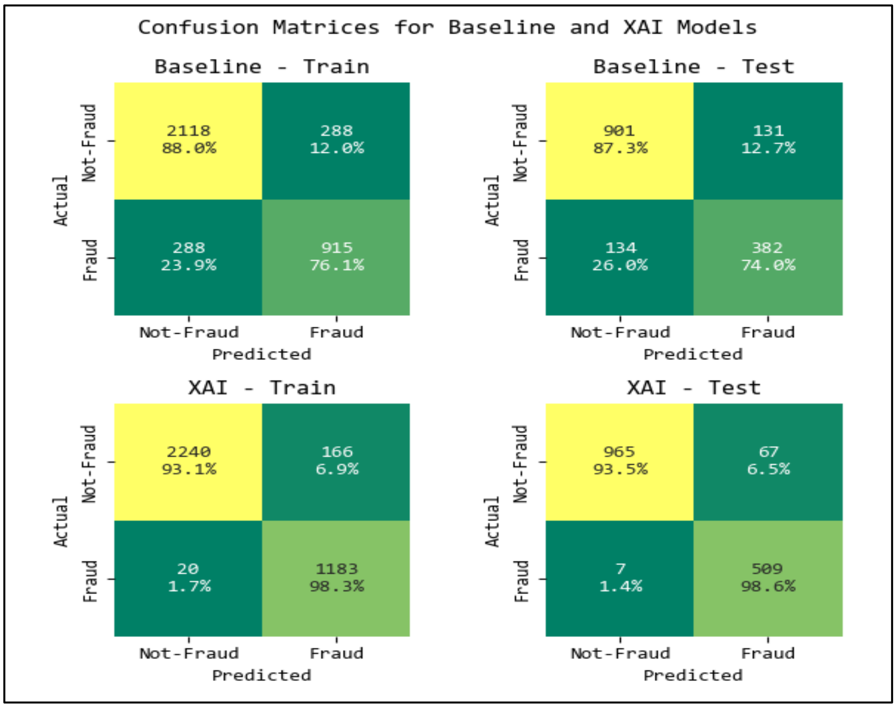

Figure 8. Confusion matrix of XAIGFL-NN technique under baseline and XAI model with IBM data

Table 3 displays the AML identification results of the XAIGFL-NN method below baseline and the XAI technique with IBM data. The outcomes imply that the XAIGFL-NN method precisely classified the samples. Below baseline with TRAP, the XAIGFL-NN approach offer result with $accur_y$ of 84.04%, $preci_n$ of 76.06%, $recal_l$ of 76.06%, $F1_{score}$ of 76.06%, MCC of 64.09%, FPR of 11.97%, ROC-AUC of 91.40%, and PR-AUC of 84.20%. In TESP, the XAIGFL-NN system produces results with $accur_y$ of 82.88%, $preci_n$ of 74.46%, $recal_l$ of 74.03%, $F1_{score}$ of 74.25%, MCC of 61.43%, FPR of 12.69%, ROC-AUC of 89.70%, and PR-AUC of 80.90%. Additionally, under XAI with TRAP, the XAIGFL-NN technique gives results with $accur_y$ of 94.85%, $preci_n$ of 87.69%, $recal_l$ of 98.34%, $F1_{score}$ of 92.71%, MCC of 89.09%, FPR of 6.90%, ROC-AUC of 99.60%, and PR-AUC of 99.20%. In TESP, the XAIGFL-NN method provides results with $accur_y$ of 95.22%, $preci_n$ of 88.37%, $recal_l$ of 98.64%, $F1_{score}$ of 93.22%, MCC of 89.87%, FPR of 6.49%, ROC-AUC of 99.20%, and PR-AUC of 98.70%.

Table 3. AML detection result of the XAIGFL-NN method with IBM data

Model	**Baseline**		**XAI**	
Split	**TRAP**	**TESP**	**TRAP**	**TESP**
Accuracy	84.04	82.88	94.85	95.22
Precision	76.06	74.46	87.69	88.37
Recall	76.06	74.03	98.34	98.64
F1-score	76.06	74.25	92.71	93.22
MCC	64.09	61.43	89.09	89.87
FPR	11.97	12.69	6.90	6.49
ROC-AUC	91.40	89.70	99.60	99.20
PR-AUC	84.20	80.90	99.20	98.70

To illustrate the enhanced performance of the XAIGFL-NN model utilizing IBM data, a short-term comparative analysis is made in Table 4 [23-25]. The results verified that the LR and KNN algorithm have display fewer classification outcomes. Meanwhile, the LSTM, TabNet, AR-GCN, and XGBoost methods have tried to achieve slightly closer classification outcomes. Moreover, the GraphSAGE technique has shown reasonable performance with $accur_y$ of 92.80%, $preci_n$ of 65.00%, $recal_l$ of 65.92%, and $F1_{score}$ of 71.00%. However, the XAIGFL-NN (XAI) approach validates promising performance with $accur_y$ of 95.22%, $preci_n$ of 88.37%, $recal_l$ of 98.64%, and $F1_{score}$ of 93.22%, where the XAIGFL-NN (Baseline) technique has less performance with $accur_y$ of 84.04%, $preci_n$ of 76.06%, $recal_l$ of 76.06%, and $F1_{score}$ of 76.06%. Based on the Computational time the proposed XAIGFL-NN (Baseline) and XAIGFL-NN (XAI) methodology obtains a less Computational time of 28.36ms and 31.94ms, compared to the other model, such as LSTM, GraphSAGE, TabNet, AR-GCN, XGBoost, LR, and KNN.

Table 4. Comparative analysis of XAIGFL-NN method with existing model using IBM data

Models	$Accur_y$	$Preci_n$	$Recal_l$	$F1_{score}$	Computation Time (ms)
LSTM	91.50	66.14	72.96	83.28	38.74
GraphSAGE	92.80	65.00	65.92	71.00	56.32
TabNet	88.00	82.91	77.53	70.55	45.87
AR-GCN	92.11	75.49	72.42	79.55	63.45
XGBoost	90.12	80.94	81.31	72.68	18.43
LR	71.00	72.85	77.06	82.47	6.91
KNN	76.00	68.00	67.92	84.16	21.62
XAIGFL-NN (Baseline)	84.04	76.06	76.06	76.06	28.36
XAIGFL-NN (XAI)	95.22	88.37	98.64	93.22	31.94

Table 5 demonstrate the computational efficiency of XAIGFL-NN model under IBM data with other techniques. Based on inference time, the XAIGFL-NN (Baseline) and XAIGFL-NN (XAI) techniques provides lesser value of 4.12ms and 4.76ms, Flops of 1.54G and 1.62G, and GPU of 1.03GB and 1.15GB. whereas the LSTM, GraphSAGE, TabNet, and AR-GCN, models attain greater time of 5.92ms, 8.45ms, 6.31ms, and 9.83ms, respectively.

Table 5. Computational efficiency of XAIGFL-NN approach under IBM data

Models	Inference Time (ms)	FLOPs (G)	GPU Memory (GB)
LSTM	5.92	1.96	1.28
GraphSAGE	8.45	3.42	1.84
TabNet	6.31	2.58	1.56
AR-GCN	9.83	4.11	2.08
XAIGFL-NN (Baseline)	4.12	1.54	1.03
XAIGFL-NN (XAI)	4.76	1.62	1.15

Table 6 indicates the ablation study of the XAIGFL-NN model. The NN, achieved an $accu_y$ of 81.76%, $prec_n$ of 72.84%, $reca_l$ of 73.18%, and $F1_{score}$ of 73.01%. and FE + NN providing an $accu_y$ of 82.93%, $prec_n$ of 74.41%, $reca_l$ of 74.86%, and $F1_{score}$ of 74.63%. Likewise, The FE + MCPS + NN, attained a better $accu_y$ to 84.04%, $prec_n$ to 76.06%, $reca_l$ to 76.06%, and $F1_{score}$ to 76.06%. Furthermore, the FE + MCPS + NN + RMSprop, additionally increased $accu_y$ to 91.68%, $prec_n$ to 85.43%, $reca_l$ to 95.31%, and $F1_{score}$ to 90.10%. At last, the XAIGFL-NN (FE + MCPS + NN + RMSprop + SHAP-GFR) model, achieved the highest outputs with an $accu_y$ of 95.22%, $prec_n$ of 88.37%, $reca_l$ of 98.64%, and $F1_{score}$ of 93.22%, emphasising the efficiency of the components.

Table 6. Ablation study of XAIGFL-NN model under IBM dataset

Models	Accuracy	Precision	Recall	F1-Score
NN	81.76	72.84	73.18	73.01
FE + NN	82.93	74.41	74.86	74.63

FE + MCPS + NN	84.04	76.06	76.06	76.06
FE + MCPS + NN + RMSprop	91.68	85.43	95.31	90.10
XAIGFL-NN (FE + MCPS + NN + RMSprop + SHAP-GFR)	95.22	88.37	98.64	93.22

Figure 9 represents the confusion matrices presented by the XAIGFL-NN method for both baseline and XAI approaches, depending on the TRAP/TESP split of PaySim data. The results indicate that the XAIGFL-NN technique effectively detects and classifies all classes accurately.

Figure 10 shows the Precision–Recall (PR) and ROC curve analyses of the XAIGFL-NN approach with PaySim data. The PR curves validate that the technique sustains higher precision when attaining strong recall over all classes, signifying efficient detection of true positives with the fewest false positives. The constant PR performance underlines the method's dependability in managing class-wise estimation. Correspondingly, the ROC curves display enhanced discrimination capability, with high true positive rates and low false positive rates across classes.

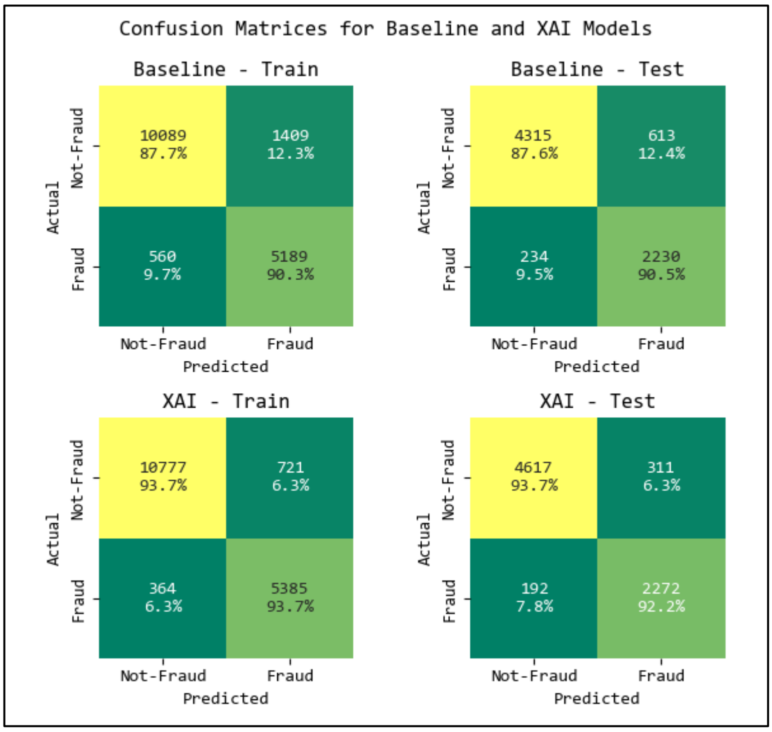

Figure 9. Confusion matrix of XAIGFL-NN technique under baseline and XAI model with PaySim data

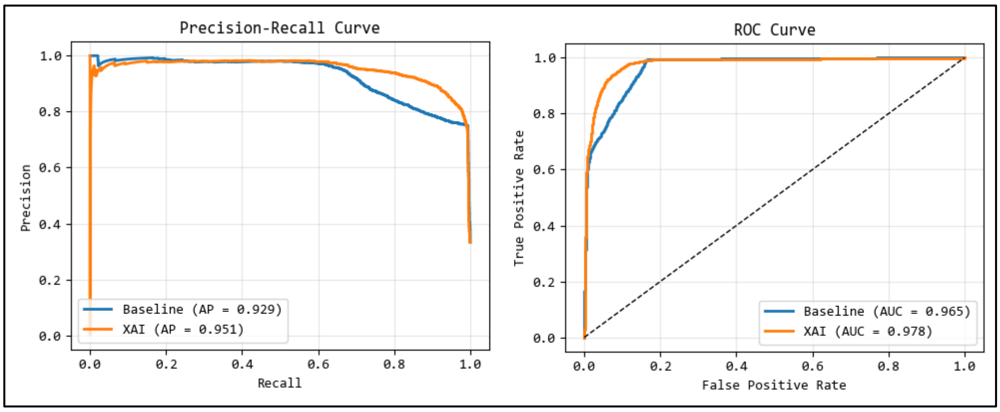

Figure 10. PR and ROC curves of the XAIGFL-NN method with PaySim data

Table 7 demonstrates the AML identification results of the XAIGFL-NN method below baseline and the XAI technique with PaySim data. The outcomes imply that the XAIGFL-NN approach precisely classified the sample. Under baseline with TRAP, the XAIGFL-NN method delivers results with $accur_y$ of 88.58%, $preci_n$ of 78.65%, $recal_l$ of 90.26%, $F1_{score}$ of 84.05%, MCC of 75.66%, FPR of 12.25%, ROC-AUC of 97.40%, and PR-AUC of 94.20%. In TESP, the XAIGFL-NN technique provides results with $accur_y$ of 88.54%, $preci_n$ of 78.44%, $recal_l$ of 90.50%, $F1_{score}$ of 84.04%, MCC of 75.64%, FPR of 12.44%, ROC-AUC of 96.50%, and PR-AUC of 92.90%. Additionally, under XAI with TRAP, the XAIGFL-NN approach gives results with $accur_y$ of 93.71%, $preci_n$ of 88.19%, $recal_l$ of 93.67%, $F1_{score}$ of 90.85%, MCC of 86.15%, FPR of 6.27%, ROC-AUC of 98.50%, and PR-AUC of 96.30%. In TESP, the XAIGFL-NN technique provide result with $accur_y$ of 93.20%, $preci_n$ of 87.96%, $recal_l$ of 92.21%, $F1_{score}$ of 90.03%, MCC of 84.93%, FPR of 6.310%, ROC-AUC of 97.80%, and PR-AUC of 95.10%.

Table 7. AML detection result of XAIGFL-NN method with PaySim data

Model	**Baseline**		**XAI**	
Split	**TRAP**	**TESP**	**TRAP**	**TESP**
Accuracy	88.58	88.54	93.71	93.2
Precision	78.65	78.44	88.19	87.96
Recall	90.26	90.5	93.67	92.21
F1-score	84.05	84.04	90.85	90.03
MCC	75.66	75.64	86.15	84.93
FPR	12.25	12.44	6.27	6.310
ROC-AUC	97.40	96.50	98.50	97.80
PR-AUC	94.20	92.90	96.30	95.10

To illustrate the enhanced performance of the XAIGFL-NN technique utilizing IBM data, a short comparative analysis is shown in Table 8. The results verified that the GraphConsis model has given fewer classification outcomes with $accur_y$ of 76.11%, $preci_n$ of 68.43%, $recal_l$ of 67.54%, and $F1_{score}$ of 66.55%. Meanwhile, the TabPFN, LGBM, and CatBoost models have tried to achieve slightly closer classification outcomes. Moreover, the Random Forest, Naïve Bayes, and FDSV technique has showed reasonable performance. Though the XAIGFL-NN (XAI) model validates promising performance with $accur_y$ of 93.71%, $preci_n$ of 88.19%, $recal_l$ of 93.67%, and $F1_{score}$ of 90.85%, where the XAIGFL-NN (Baseline) approach has less performance with $accur_y$ of 88.58%, $preci_n$ of 78.65%, $recal_l$ of 90.26%, and $F1_{score}$ of 84.05%. Based on the Computational time the proposed XAIGFL-NN (Baseline) and XAIGFL-NN (XAI) methodology obtains a less Computational time of 28.36ms and 31.94ms, compared to the other model, such as LSTM, GraphSAGE, TabNet, AR-GCN, XGBoost, LR, and KNN.

Table 8. Comparison result of XAIGFL-NN method with existing model using PaySim data

Models	$\boldsymbol{Accur_y}$	$\boldsymbol{Preci_n}$	$\boldsymbol{Recal_l}$	$\boldsymbol{F1_{score}}$	**Computation Time (ms)**
Random Forest	90.00	78.04	83.06	76.58	19.84
Naïve Bayes	91.00	83.84	79.25	77.52	5.43
TabPFN	86.00	65.71	78.67	65.16	47.36
LGBM	89.00	72.07	73.34	78.96	15.27
FDSV	91.09	66.71	71.79	84.67	53.74
CatBoost	88.45	71.01	73.86	79.46	22.15
GraphConsis	76.11	68.43	67.54	66.55	61.28
XAIGFL-NN (Baseline)	88.58	78.65	90.26	84.05	28.36
XAIGFL-NN (XAI)	93.71	88.19	93.67	90.85	31.94

The proposed XAIGFL-NN architecture efficiently improves anti-money laundering detection by combining field-aware balanced sampling, explainable neural learning, and feature engineering. The use of a neural network optimized with RMSprop empowers effective learning of discriminatory transaction patterns in highly imbalanced financial datasets. In addition, SHAP-based explainability not only delivers interpretability but also guides feature refinement, enhancing both detection accuracy and transparency. Experimental outcomes on IBM AMLSim and PaySim datasets signify that the proposed system outperforms conventional techniques. Therefore, the architecture delivers strong and interpretable results to detect suspicious financial activities in larger-scale settings.

Table 9 illustrates the computational efficiency of the XAIGFL-NN model under PaySim data compared with existing models. In accordance with inference time, the XAIGFL-NN (Baseline) and XAIGFL-NN (XAI) techniques provide lesser values of 4.12ms and 4.76ms, FLOPs of 1.54G and 1.62G, and GPU usage of 1.03GB and 1.15GB. In contrast, the TabPFN, FDSV, and GraphConsis models attain greater times of 6.58ms, 7.91ms, and 8.96ms, respectively.

Table 9. Computational efficiency of XAIGFL-NN method under PaySim data

Models	Inference Time (ms)	FLOPs (G)	GPU Memory (GB)
TabPFN	6.58	2.73	1.63
FDSV	7.91	3.24	1.92
GraphConsis	8.96	3.86	2.17
XAIGFL-NN (Baseline)	4.12	1.54	1.03
XAIGFL-NN (XAI)	4.76	1.62	1.15

Table 10 represents the ablation study of the XAIGFL-NN model. The NN, achieved an $accu_y$ of 86.74%, $prec_n$ of 75.82%, $reca_l$ of 88.41%, and $F1_{score}$ of 81.63%. and FE + NN providing an $accu_y$ of 87.63%, $prec_n$ of 77.08%, $reca_l$ of 89.18%, and $F1_{score}$ of 82.67%. Meantime, The FE + MCPS + NN, attained a better $accu_y$ to 88.58%, $prec_n$ to 78.65%, $reca_l$ to 90.26%, and $F1_{score}$ to 84.05%. Furthermore, the FE + MCPS + NN + RMSprop, additionally increased $accu_y$ to 91.92%, $prec_n$ to 85.74%, $reca_l$ to 92.03%, and $F1_{score}$ to 88.77%. At last, the XAIGFL-NN (FE + MCPS + NN + RMSprop + SHAP-GFR) model, achieved the highest outputs with an $accu_y$ of 93.71%, $prec_n$ of 88.19%, $reca_l$ of 93.67%, and $F1_{score}$ of 90.85%, emphasising the efficiency of the components.

Table 10. Ablation study of XAIGFL-NN method using PaySim data

Models	Accuracy	Precision	Recall	F1-Score
NN	86.74	75.82	88.41	81.63
FE + NN	87.63	77.08	89.18	82.67
FE + MCPS + NN	88.58	78.65	90.26	84.05
FE + MCPS + NN + RMSprop	91.92	85.74	92.03	88.77
XAIGFL-NN (FE + MCPS + NN + RMSprop + SHAP-GFR)	93.71	88.19	93.67	90.85

5. Conclusion and Future Works

In this paper, the XAIGFL-NN framework is proposed as an effective Anti-Money Laundering (AML) solution. It applies the SHAP method not only for interpretation but also to highlight important features in order to increase the interpretability of the model and its performance. The framework is trained on the benchmark data IBM AMLSim and PaySim, both of which have imbalanced class distribution, using a minority-class preserving undersampling technique. Domain-based feature extraction is applied, resulting in the use of a Fully Connected Neural Network (FCNN) with RMSprop optimization as the base classifier. Even though the results seem to be promising, the framework was only validated empirically with no regard to real-world conditions such as inference time and scalability. Additionally, even though the SHAP method makes the framework interpretable, it requires increased computational resources. Future research needs to address practical efficiency issues, reduce computational expenses, apply continual learning for ever-changing fraud patterns, and maybe even use graph learning and attention-based neural networks in order to make the framework efficient in combating sophisticated money laundering schemes.

Declarations

Data Availability Statement

The data that support the findings of this study are openly available in Kaggle repository at https://www.kaggle.com/datasets/anshankul/ibm-amlsim-example-dataset and https://www.kaggle.com/datasets/ealaxi/paysim1, reference number [16, 17].

Code Availability

The source code of this study is available at https://github.com/zgenz1537-code/AMLFraudDetection.git.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript. All authors contributed to the manuscript and approved the final version.

References

[1] Amoako, Elizabeth Kuukua Woode, Victor Boateng, Ola Ajay, Tobias Kwame Adukpo, and Nicholas Mensah. "Exploring the Role of Machine Learning and Deep Learning in Anti-Money Laundering (AML) Strategies Within US Financial Industry: A Systematic Review of Implementation, Effectiveness, and Challenges." Finance & Accounting Research Journal 7, no. 1 (2025): 22-36.

[2] Bi, Wenyu, Toan Khang Trinh, and Shukai Fan. "Machine Learning-based Pattern Recognition for Anti-Money Laundering in Banking Systems." Journal of Advanced Computing Systems 4, no. 11 (2024): 30-41.

[3] Pazos, Jorge Felix Martínez, Jorge Gulín González, David Batard Lorenzo, and Jorge Alejandro Robaina Morales. "Fraud Transaction Detection for Anti-Money Laundering Systems based on Deep Learning." Journal of Emerging Computer Technologies 3, no. 1 (2024): 29-34.

[4] Bakry, Ahmed N., Almohammady S. Alsharkawy, Mohamed S. Farag, and Kamal R. Raslan. "Automatic Suppression of False Positive Alerts in Anti-Money Laundering

Systems Using Machine Learning: AN Bakry et al." The Journal of Supercomputing 80, no. 5 (2024): 6264-6284.

[5] Konstantinidis, Giannis, and Alexander Gegov. "Deep Neural Networks for Anti Money Laundering Using Explainable Artificial Intelligence." In 2024 IEEE 12th international conference on intelligent systems (IS), IEEE, 2024, 1-6.

[6] Iguodala, O., and Aghogho Oyiborhoro. "AI-Powered Anti-Money Laundering (AML) and Fraud Detection-Enhancing Financial Security Through Intelligent Fraud Detection." World Journal of Advanced Research and Reviews 26 (2025): 3702-3714.

[7] Oztas, Berkan, Deniz Cetinkaya, Festus Adedoyin, Marcin Budka, Gokhan Aksu, and Huseyin Dogan. "Transaction Monitoring in Anti-Money Laundering: A Qualitative Analysis and Points of View from Industry." Future Generation Computer Systems 159 (2024): 161-171.

[8] Olaniyi, Oluwadayo Mafolasere, Ifesinachi Stephen Aroh, Henry Onyii, Olufunke Cynthia Metibemu, and Oluwaseun Ibrahim Akinola. "Graph Neural Networks for Multi-Layered Financial Crime Network Detection: An Explainable AI Framework for Anti-Money Laundering." Journal of Engineering Research and Reports 28, no. 2 (2026): 18-36.

[9] Lin, Ziqian, Qining Luo, Dongze Wu, Jie Shen, Lingying Li, Xunyi Nong, and Zhenkai Qin. "Detecting Illicit Transactions in Bitcoin: A Wavelet-Temporal Graph Transformer Approach for Anti-Money Laundering." Scientific Reports 16, no. 1 (2026): 1548.

[10] Sidiq, Mashkhal Abdalwahid, and Yimamu Kirubel Wondaferew. "Anti-Money Laundering Systems Using Deep Learning." arXiv preprint arXiv:2509.19359 (2025).

[11] Wang, Qianyu, Wei-Tek Tsai, and Bowen Du. "RMGANets: Reinforcement Learning-Enhanced Multi-Relational Attention Graph-Aware Network for Anti-Money Laundering Detection." Complex & Intelligent Systems 11, no. 1 (2025): 5.

[12] Wu, You, Ying Qin, Xin Su, and Yuxiu Lin. "Transformer-based Risk Monitoring for Anti-Money Laundering with Transaction Graph Integration." In Proceedings of the 2025 2nd International Conference on Digital Economy, Blockchain and Artificial Intelligence, 2025, 388-393.

[13] Yu, Qian, Zhen Xu, and Zong Ke. "Deep Learning for Cross-Border Transaction Anomaly Detection in Anti-Money Laundering Systems." In 2024 6th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), IEEE, 2024, 244-248.

[14] Wan, Fei, and Ping Li. "A Novel Money Laundering Prediction Model based on a Dynamic Graph Convolutional Neural Network and Long Short-Term Memory." Symmetry 16, no. 3 (2024): 378.

[15] Rajath Karangara, Saikiran Subbagari, "Filtering and Detection of Anti Money Laundering with the Aid of Optimization-Enabled Machine Learning Techniques," International Journal of Computer & Organization Trends (IJCOT), vol. 14, no. 1, 2024, 1-11.

[16] https://www.kaggle.com/datasets/anshankul/ibm-amlsim-example-dataset

[17] https://www.kaggle.com/datasets/ealaxi/paysim1

[18] Zhou, Wensheng, Chen Liu, Peng Yuan, and Lei Jiang. "An Undersampling Method Approaching the Ideal Classification Boundary for Imbalance Problems." Applied Sciences 14, no. 13 (2024): 5421.

[19] Haq, Hafiz Burhan Ul, Rabia Younis, and Muhammad Shujat Ali. "Towards Robust Network Security: Evaluating Machine Learning Algorithms for Intrusion Detection." Decision Making Advances 3, no. 1 (2025): 126-138.

[20] Björnsson, Ola, and Per Tunestål. "Knock Detection with Ion Current and Vibration Sensor: A Comparative Study of Logistic Regression and Neural Networks." Energies 17, no. 22 (2024): 5693.

[21] Mondal, Subhash, Ranjan Maity, and Amitava Nag. "An Efficient Artificial Neural Network-Based Optimization Techniques for the Early Prediction of Coronary Heart Disease: Comprehensive Analysis." Scientific Reports 15, no. 1 (2025): 4827.

[22] Zhu, Wenbing, Shihan Luo, Hyowon Lee, Amandeep Singh, and Siby Samuel. "Lightweight and Interpretable Transformer-Based Models for Predicting Driving Comfort in Autonomous Vehicles." IEEE Access 14 (2026): 9435-9452.

[23] RAJA, MAMUNUR R., Md Anwar Hosen, Md Farhad Kabir, Sharmin Sultana, Shah Ahammadullah Ashraf, and Rakibul Islam. "Detecting and Preventing Money Laundering Using Deep Learning and Graph Analysis." International Journal of Advanced Computer Science & Applications 16, no. 6 (2025).

[24] Li, Xiang, Lei Chu, Yujun Li, Zhanjun Xing, Fengqian Ding, Jintao Li, and Ben Ma. "An Intelligent Financial Fraud Detection Support System Based on Three-Level Relationship Penetration." Mathematics 12, no. 14 (2024): 2195.

[25] Yang, Yiwen, Chengjun Xu, and Guisheng Tian. "Lightweight Financial Fraud Detection Using a Symmetrical GAN-CNN Fusion Architecture." Symmetry 17, no. 8 (2025): 1366.