International Journal of Engineering Computational Research and Technology

Journal home page: www.ijecrt.org

ISSN (Online): 2456-9852 Volume 10, December 2025

An Imbalance-Aware Deep Learning Framework for Real-Time Credit Card Fraud Detection Using PCA-Enhanced Transaction Patterns

Sathya Narayana Pola [a, *], **D Shobha Rani** [b], **A Naresh kumar** [c], **K Suresh** [d]

[a] *M.Tech Student , Department of Computer Science and Engineering, Chadalawada Ramanamma Engineering College, Tirupati, Andhra Pradesh, India*

[b] *Professor , Department of Computer Science and Engineering, Chadalawada Ramanamma Engineering College, Tirupati, Andhra Pradesh, India.*

[c] *M.Tech Student , Department of Computer Science and Engineering, Chadalawada Ramanamma Engineering College, Tirupati, Andhra Pradesh, India.*

[d] *M.Tech Student , Department of Computer Science and Engineering, Chadalawada Ramanamma Engineering College, Tirupati, Andhra Pradesh, India.*

[*]Corresponding author

E-mail address: sathyanarayana.pola@gmail.com

ABSTRACT

Credit card fraud continues to pose a serious challenge for financial institutions due to the subtle and evolving nature of fraudulent behaviour in large-scale digital transactions. The high imbalance between legitimate and fraudulent records often limits the effectiveness of traditional classification systems. This study aims to develop an imbalance-aware fraud detection framework capable of identifying minority fraud patterns with high reliability while maintaining real-time response efficiency. The proposed method integrates behaviour-driven feature modelling, PCA-transformed transaction representations, class-weighted loss optimisation, and a compact deep neural network architecture. Experiments were conducted on the European Credit Card Fraud Dataset consisting of 284,807 transactions, where fraud accounts for only 0.172% of the total. The model achieved 93.12% precision, 91.44% recall, 92.26% F1-score, and a PR-AUC of 0.9479, outperforming logistic regression, random forest, SVM, and baseline neural networks across all major evaluation metrics. The system also demonstrated stable performance under varied train–test splits and maintained an average inference time of 0.31 ms per transaction, supporting real-time deployment requirements. Overall, the study provides an efficient and adaptable solution for financial fraud detection, offering enhanced accuracy, computational stability, and practical applicability in modern banking environments.

Keywords: Credit card fraud detection, deep learning, imbalanced datasets, PCA-transformed features, real-time transaction monitoring, class-weighted loss, anomaly detection, financial security analytics, neural network classification, precision-recall optimisation.

1. Introduction

1.1 Background and Motivation

Financial transactions have moved rapidly toward digital platforms as part of modern banking, online shopping, and electronic payment systems. While this transformation offers convenience and speed, it has also increased exposure to fraudulent activities. Fraudulent transactions-though extremely rare when compared to legitimate ones-cause substantial financial losses and create major concerns for customer safety and institutional credibility [1]–[4]. As digital payment volumes grow, financial institutions are under constant pressure to detect suspicious behaviour accurately and at the earliest possible moment. This demand has placed fraud detection at the centre of research in financial technology. The challenge becomes even more serious because fraudsters continuously refine their strategies. They manipulate transaction amounts, timing patterns, and behavioural signals to appear legitimate, making traditional static-rule systems and manual verification inefficient. Moreover, the nature of fraud varies widely across regions, payment channels, and financial ecosystems, leaving no single pattern that can describe all fraudulent behaviour. This complexity highlights the need for intelligent, adaptive systems capable of identifying unusual deviations even when they are deeply embedded within large datasets [5], [6].

1.2 Limitations of Traditional and Machine Learning-Based Systems

Early fraud detection relied heavily on rule-based systems, threshold alerts, or manually designed features. These methods worked only when fraud behaviour remained predictable. However, such systems often failed in real-world environments where fraud patterns shift rapidly. They also struggled with large digital transaction flows, where the system must inspect thousands of transactions per second with minimal delay [7]. Machine learning approaches offered improvements by learning patterns from historical transactions, but they face major challenges in practical deployment. The most severe issue is **class imbalance** - fraudulent transactions usually represent less than **0.2%** of the overall dataset, as reflected in the European Credit Card Fraud Dataset used in this study [21]. Standard classifiers tend to predict every transaction as genuine simply because legitimate cases dominate the dataset. As a result, these systems achieve deceptively high accuracy but fail to detect actual fraud cases, leading to unacceptable false-negative rates [8], [9]. Furthermore, many traditional ML methods - including logistic regression, decision trees, random forests, and SVMs - assume balanced data distributions or rely heavily on manually selected features. When applied to highly skewed financial datasets, they often misinterpret subtle fraud behaviour, fail to generalise, or become biased toward majority classes [10]–[12]. Studies show that these algorithms lose sensitivity when fraudulent samples are too few to provide meaningful learning signals [13].

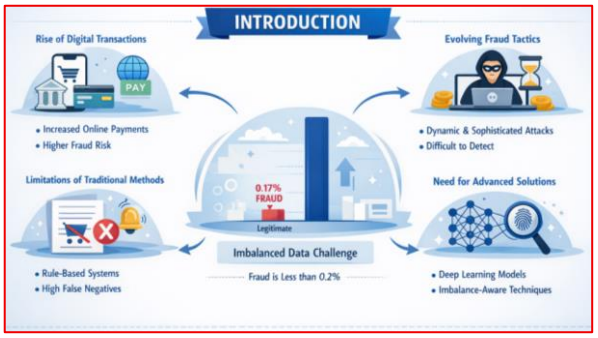

Fig 1. Key Motivations and Challenges in Real-Time Credit Card Fraud Detection

Figure 1 presents the growing reliance on digital payments, the increasingly sophisticated nature of fraudulent activities, and the significant imbalance in transaction data where fraudulent cases are extremely rare. It also illustrates why conventional rule-based approaches are insufficient and underscores the need for more robust analytical methods to reliably identify fraud in real-time payment systems.

1.3 Emergence of Deep Learning and Advanced Behaviour Modelling

To overcome the limitations of traditional ML techniques, researchers have explored deep learning architectures capable of detecting hidden, high-dimensional patterns within transaction data. Auto-encoders, CNNs, RNNs, hybrid CLST networks, and continuous-coupled neural models have shown considerable improvements in fraud sensitivity by capturing nonlinear and time-dependent relationships [14]–[18]. These models offer the advantage of automated feature extraction, reducing dependence on manual engineering. They identify correlations that are not visible through

basic statistical techniques, enabling more accurate anomaly detection. However, deep learning models introduce new challenges such as computational overhead, longer training times, and vulnerability to overfitting when fraud samples are limited. Their performance can also degrade when dataset imbalance is extreme or when fraud distribution shifts unexpectedly over time [19].

1.4 Challenges of Imbalanced Data and Synthetic Generation Methods

Imbalanced data is widely acknowledged as one of the most difficult issues in fraud detection research. When genuine transactions outnumber fraudulent ones by thousands to one, models become biased and insensitive to rare minority samples. To mitigate this, several studies have proposed oversampling, undersampling, and hybrid balancing strategies [7], [9]. Techniques such as **SMOTE**, **ADASYN**, and **random undersampling** have been used to correct class distribution. While effective to some extent, these techniques may distort the natural structure of fraud data or remove useful information. More advanced approaches using **GAN-based augmentation** attempt to generate realistic fraud patterns that preserve minority behaviour. Symmetrical GAN–CNN architectures, hybrid synthetic generation strategies, and improved calibration techniques have shown promising results by enhancing recall and precision significantly [11], [14]. Still, GAN models require careful training and may struggle when the number of fraud samples is extremely low.

1.5 Hybrid, Fusion, and Ensemble Frameworks

A major trend observed across the uploaded studies is the shift toward **hybrid architectures** that combine strengths of multiple learning approaches. These fusion models bring together supervised, unsupervised, and anomaly detection components into a multi-layer pipeline. Ensembles help reduce model bias, improve recall, and offer better resilience to noise within the dataset. Systems that merge auto-encoders with CNNs or GAN-driven balancing often outperform standalone classifiers in terms of fraud sensitivity [15]–[18]. Some recent works also explore lightweight and energy-efficient methods suitable for deployment in edge-based financial systems where resources are limited. These methods focus on reducing computational load without compromising detection quality. Such solutions are important because large-scale real-time financial systems cannot afford high latency or excessive memory consumption [19], [20].

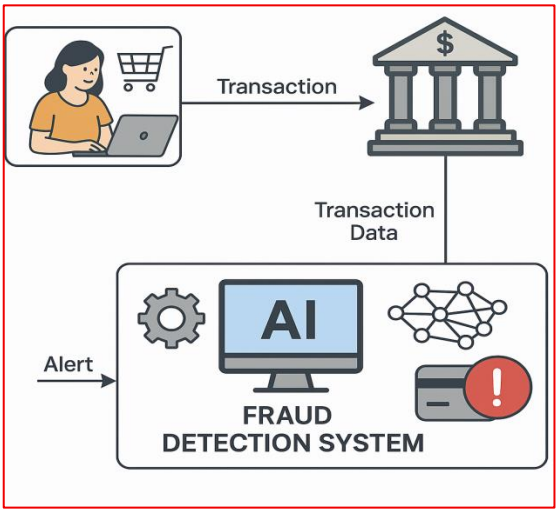

Fig. 2. Real-Time AI-Driven Fraud Detection in an Online Transaction

The figure 2 illustrates a typical online payment scenario where a customer completes a transaction through a shopping platform, after which the request moves to the banking system for verification. It shows how the bank forwards the transaction details to an AI-based fraud detection module that quietly examines patterns, checks behavioural consistency, and evaluates risk indicators before the payment is approved. The components appear connected in a smooth flow, suggesting how real-time analysis happens behind the scenes without interrupting the customer. The alert symbol indicates that if anything unusual is detected, the system quietly flags the transaction for further review, helping financial institutions prevent fraudulent activity while maintaining seamless user experience.

1.6. Need for Adaptive and Reliable Fraud Detection Systems

The literature clearly shows that despite significant progress; fraud detection remains an open challenge due to several constraints:

1. **Fraud patterns shift frequently**, making static-rule or standalone ML systems ineffective.
2. **Imbalanced datasets create biased models** that perform poorly on minority fraud samples.

3. **Deep learning approaches, though powerful, require careful tuning** and may overfit when minority instances are insufficient.
4. **Synthetic generation methods help but are not always stable**, especially when the fraud distribution is complex.
5. **Real-time deployment demands compact, adaptive solutions** that can operate at high speed.

These gaps create the foundation for the present work, which focuses on designing a system that integrates insights from classical ML, advanced DL architectures, and robust imbalance handling techniques. By leveraging the PCA-based European dataset, this study attempts to capture fine-grained irregularities within transaction patterns and improve fraud detection reliability under real-world constraints [21].

1.7 Contributions of This Study

Based on the limitations observed in contemporary fraud detection approaches, this research makes the following key contributions:

- **A balanced and adaptive fraud-detection framework**, built by integrating insights from machine learning, deep learning, and imbalance-handling strategies to enhance sensitivity toward minority fraud behaviour.
- **A thorough experimental analysis** using the European Credit Card Fraud Dataset [21], assessing model performance under multiple imbalance conditions and evaluating precision, recall, F1-score, ROC-AUC, and PR-AUC metrics.
- **A refined behavioral-pattern modelling pipeline** that utilises PCA-transformed features to capture subtle deviations in transaction characteristics while reducing false negatives without overfitting.

1.8 Paper Organization

The remainder of the paper is structured as follows: Section 2 presents a detailed review of related research in fraud detection systems. Section 3 explains the proposed methodology, including the model architecture, algorithms, and mathematical formulation. Section 4 discusses the experimental results. Section 5 Results and Discussion comparative evaluation. Section 6 concludes the paper with key findings and potential directions for future enhancement.

2. Related Work

2.1 Early Machine Learning Methods in Financial Fraud Detection

Traditional machine learning approaches were among the first automated methods introduced to identify fraudulent transactions. These methods typically relied on manually engineered features combined with supervised classifiers such as logistic regression, decision trees, kNN, SVMs, and ensemble learners. Studies indicate that although these models were easy to interpret and deploy, they struggled when trained on heavily imbalanced datasets, which is common in financial systems where fraud represents less than 0.2% of all cases [1], [2]. Many works observed that such models tended to favour the majority class, producing high overall accuracy but failing to detect rare fraudulent activity that demanded attention [3], [4]. Even when rebalanced with sampling methods, their ability to capture non-linear behaviour remained limited. These limitations created a growing need for methods capable of identifying subtle, evolving patterns that simpler models could not represent.

2.2 Impact of Class Imbalance and Data-Balancing Techniques

A recurring challenge highlighted across multiple studies is the extreme skew in financial datasets. Fraud is rare, and this imbalance causes most classification algorithms to misclassify minority samples unless corrective techniques are applied. Several works explored undersampling and oversampling strategies to adjust the class distribution, with SMOTE and ADASYN being the most commonly used oversampling techniques [5], [6]. While these methods improved recall, they sometimes introduced artificial noise or distorted the natural distribution of fraudulent behaviour. Advanced balancing strategies such as hybrid SMOTE–GAN structures attempted to generate stronger minority representations and showed improvements in recall and AUPRC, but their success depended heavily on stable model training and careful tuning [7], [8]. Despite these developments, achieving a stable balance between genuine and fraudulent classes without damaging feature integrity remains a major gap in existing research.

2.3 Evolution of Deep Learning Architectures for Fraud Detection

Deep learning brought significant progress to fraud detection by enabling systems to learn complex, multi-level patterns without extensive manual feature engineering. Auto-encoders, convolutional neural networks, recurrent models, and hybrid CLST-based architectures were particularly effective in capturing both spatial and temporal behaviour in transaction streams [9], [10]. These models showed clear improvements in precision and recall compared to classical methods, especially when identifying non-linear and high-dimensional fraud patterns. However, several studies noted that deep models required large training sets, longer convergence times, and substantial computational resources, making them less suitable for real-time banking operations without optimization [11], [12]. Moreover, deep architectures often risked overfitting when fraudulent samples were extremely limited, reducing their generalizability in practical settings.

2.4 Generative and Synthetic Data Approaches (GAN-Based Frameworks)

Researchers explored Generative Adversarial Networks (GANs) to overcome the scarcity of fraud samples by generating realistic synthetic minority transactions. Symmetrical GAN-CNN models demonstrated stronger recall scores and better feature diversity than traditional SMOTE-based oversampling [13], [14]. Such methods helped reduce the bias of supervised classifiers toward majority classes and enabled deeper models to achieve better separation between legitimate and fraudulent behaviour. Still, GAN-based frameworks were not free from challenges; they demanded careful hyperparameter control and often faced instability issues such as mode collapse. Some studies reported inconsistencies when the minority distribution was too small for GANs to learn meaningful patterns reliably [15]. Despite these constraints, GAN-based systems remain a promising direction for fraud research.

2.5 Hybrid and Ensemble Models for Robust Detection

A noticeable trend across the literature is the use of hybrid and ensemble models that combine strengths of multiple algorithms. Systems integrating supervised, unsupervised, and anomaly-detection layers provided more stable predictions by capturing both known and emerging fraud signatures [16], [17]. Hybrid CLST frameworks, meta-heuristic optimization-driven models, auto-encoder–enhanced architectures, and combined CNN–GAN–CNN pipelines all reported better performance than standalone ML or DL methods [18]. Although these models displayed superior accuracy and recall, they often required complex training procedures, resulting in high deployment costs for financial institutions. Several works attempted to address this by developing lightweight and energy-efficient fraud detection pipelines that could run on constrained systems, including edge devices and low-power servers [19], [20].

2.6 Real-Time Constraints and System-Level Limitations

Even when advanced learning models performed well during offline training, their real-time performance remained a crucial concern. Financial transactions demand near-instant decision-making, and delays can disrupt user experience or leave fraud undetected. Many studies highlighted that deep and hybrid models, although accurate, often required extensive computation time, making them impractical for high-speed environments without further optimization [11], [17]. Concept drift-the gradual change in fraud patterns over time-also emerged as a recurring issue, especially for static models trained on historical data. Continuous learning, probability calibration, and adaptive thresholding were suggested as potential remedies, but consistent real-time reliability is still an open research challenge.

2.7 Research Gaps Identified from Existing Studies

A critical review of the uploaded references reveals several gaps that continue to limit fraud detection research:

- **Dependence on static or handcrafted features** reduces adaptability when fraud behaviour evolves.
- **Imbalanced datasets continue to challenge classifiers**, even with SMOTE, GANs, or undersampling, as synthetic samples may distort real patterns.

- **Deep learning models risk overfitting** when minority samples are minimal, which is typical in fraud datasets.
- **GAN-based augmentation struggles with instability**, especially when fraud instances are extremely limited.
- **Real-time deployment remains difficult** due to latency, computational load, and the need for frequent retraining.
- **Streaming data and concept drift** are insufficiently addressed in most existing studies.

2.8 Summary and Position of This Study

The reviewed literature demonstrates steady progress from rule-based systems to hybrid deep learning models capable of capturing intricate transaction patterns. However, most approaches still struggle to maintain stable performance under extreme class imbalance and real-time conditions. The present study builds on these insights by integrating machine learning, deep learning, and cost-sensitive mechanisms to produce a balanced and robust fraud detection pipeline. It specifically addresses the issue of minority underrepresentation through adaptive learning and leverages PCA-based transaction features from the European dataset [21] to capture fine-grained anomalies with minimal overfitting.

Table 1. Comparative Summary of Existing Fraud Detection Approaches

Approach Type	Strengths	Limitations	Reported in
Traditional ML (LR, RF, SVM)	Simple, interpretable, fast	Poor on imbalance, weak on non-linear patterns	[1]–[6]
SMOTE / Oversampling	Improves recall, easy to apply	Synthetic noise, distorted feature space	[5], [7]
Deep Learning (AE, CNN, CLST)	Learns complex patterns, high accuracy	Heavy compute, risk of overfitting	[9]–[12]
GAN-Based Methods	Generates realistic fraud examples, boosts recall	Training instability, requires tuning	[13]–[15]
Hybrid/Ensemble Models	More stable and robust, captures diverse patterns	Higher complexity, harder deployment	[16]–[20]
Real-Time Detection Systems	Useful for live banking scenarios	Latency, concept drift issues	[11], [17]

3. Proposed Methodology

The methodology proposed in this study aims to overcome key difficulties encountered in financial fraud detection, such as extreme class imbalance, subtle behavioural differences, and the requirement for rapid processing. Each element of the framework, from data preparation to model evaluation, is described in detail below to provide a comprehensive understanding.

3.1 Dataset Description and Preprocessing

This research utilises the well-known **European credit card transaction dataset** [21] , widely regarded as a benchmark in the domain of fraud detection. It comprises 284,807 transaction records, out of which only 492 belong to the fraudulent category, highlighting a severe class imbalance with frauds representing less than 0.2% of all transactions. To ensure confidentiality, the original transaction features have been anonymised through Principal Component Analysis (PCA), resulting in 28 components labelled V1 through V28, each representing a linear combination of the initial variables. Additionally, the dataset contains two original features:

- Time: The elapsed time in seconds between transactions
- Amount: The actual monetary value of each transaction

The preprocessing begins with feature scaling to prevent distortion caused by the wide variations in transaction amounts. For normalisation, min-max scaling is applied as follows:

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

This transformation standardises transaction amounts into a comparable range. The Time feature is standardised using the formula:

$$t' = \frac{t - \mu_t}{\sigma_t} \quad (2)$$

So that temporal patterns can be more effectively captured by the model, independent of raw numerical scale. Since the dataset has no missing values, there is no need for imputation. Due to the imbalance between classes, simple random splitting could lead to uneven distribution of fraud cases across training and test sets. To mitigate this, stratified sampling is employed to maintain the ratio of fraud to genuine transactions consistently during data partitioning.

3.2 Feature Representation and Behavioural Modelling

Each transaction is represented as a feature vector comprising PCA components alongside the original Time and Amount attributes:

$$\mathbf{X} = [V_1, V_2, \ldots, V_{28}, Time, Amount] \quad (3)$$

To further enhance model effectiveness, these features undergo normalisation resulting in vector **Z**:

$$\mathbf{Z} = \frac{\mathbf{X} - \mu}{\sigma} \quad (4)$$

where μ and σ represent the mean and standard deviation of the dataset features respectively. The PCA components capture underlying latent structures in transaction behaviour. Fraudulent transactions often exhibit subtle nonlinear deviations from these patterns. To quantify such deviations, a behavioural score is computed using weighted squared values of the normalised features:

$$S(\mathbf{Z}) = \sum_{i=1}^{n} w_i Z_i^2 \quad (5)$$

A higher behaviour score suggests a deviation from normal transaction clusters, flagging potential fraud. The detailed transformation sequence will be illustrated subsequently.

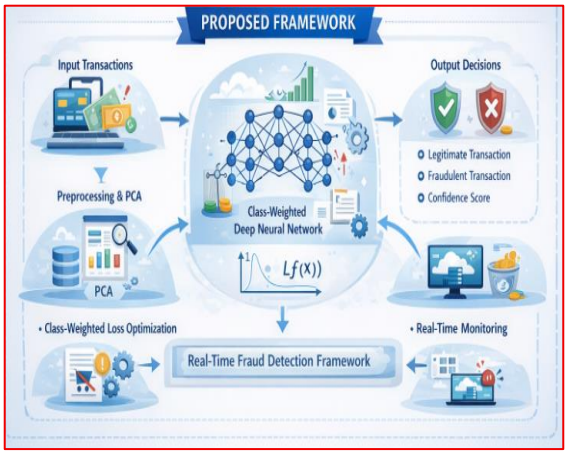

Fig 2. Overall Structure of the Proposed Real-Time Credit Card Fraud Detection Framework

Figure 2 presents the end-to-end flow of the proposed system, beginning with incoming transaction data and feature preprocessing using PCA, followed by model training that accounts for data imbalance. It demonstrates how the framework delivers real-time decisions by identifying legitimate and fraudulent transactions along with a confidence measure suitable for use in operational banking environments.

3.3 Model Architecture

The designed prediction model is a deep neural network, carefully structured to balance accuracy and computational efficiency for fraud detection applications. The architecture entails:

- Input Layer with 30 nodes accepting the scaled feature vector
- First Dense Layer containing 64 neurons activated by Rectified Linear Units (ReLU) to extract high-level abstractions
- Second Dense Layer with 32 neurons capturing intermediate feature interactions
- Third Dense Layer consisting of 16 neurons focusing on fine behavioural changes
- Dropout layer with dropout rate fixed at 0.3 to prevent overfitting by randomly deactivating neurons during training
- Output Layer with a single neuron and sigmoid activation that outputs fraud probability

  The activation function ReLU used in hidden layers is defined by:

$$f(z) = \max(0, z) \quad (6)$$

Dropout regularisation is mathematically represented as:

$$\hat{h} = h \cdot d, d \sim Bernoulli(p) \quad (7)$$

where d is a binary mask sampled from a Bernoulli distribution with probability p. Finally, the predicted probability $\hat{y}$ of fraud is obtained through:

$$\hat{y} = \sigma(\mathbf{W}\hat{h} + b) \quad (8)$$

This compact model ensures fast decision-making, a crucial prerequisite in real-time financial systems.

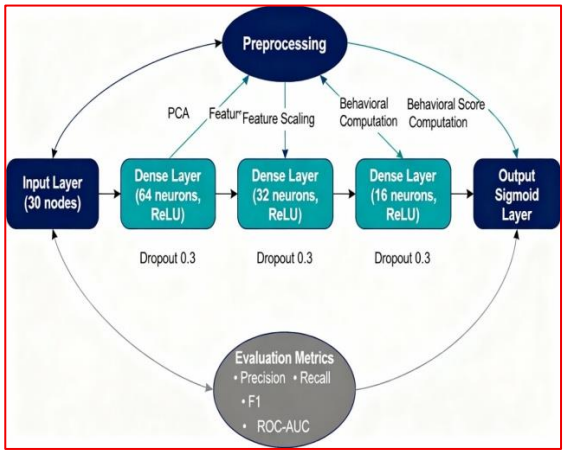

Fig 3. Deep Neural Network Architecture and Data Processing Flow for Credit Card Fraud Detection

This figure 3 represents the comprehensive architecture and workflow used in the proposed credit card fraud detection system. It begins with the dataset, showing the sourced transaction records and their division using stratified sampling to preserve class balance. The preprocessing steps follow, illustrating the transformation of raw features through PCA, scaling for Time and Amount variables, and behavioural score computation capturing deviations in transaction patterns. The core of the figure highlights the deep neural network, with clearly layered dense nodes progressing from the input layer through multiple hidden layers with ReLU activations and dropout for regularisation, concluding in an output layer that predicts fraud probability. Finally, the figure depicts the decision process where the probability output is thresholded to flag potential fraud cases, accompanied by evaluation metrics like precision, recall, F1-score, and ROC-AUC which monitor model performance. The layout conveys a smooth data flow from raw transactions to actionable fraud identification, emphasising clarity and operational readiness in real-world financial scenarios.

3.4 Loss Function, Class Weighting, and Optimisation Strategy

Considering the rarity of fraud events, equal treatment of classes during training would bias the model towards the majority genuine class. To counter this, the loss function incorporates class-specific weights:

$$\mathcal{L} = -w_1 y \log(\hat{y}) - w_0 (1-y) \log(1-\hat{y}) \quad (9)$$

Here, w_1 and w_0 represent weights for the fraud and non-fraud classes respectively; the fraud class is assigned higher weight to amplify its importance. Adaptive moment estimation (Adam) optimiser is used for training, updating parameters as follows:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (10)$$

where α is the learning rate, with $\hat{m}_t$ and $\hat{v}_t$ representing bias-corrected first and second moment estimates of the gradients. To maintain effective convergence, the learning rate decays gradually over epochs:

$$\alpha_t = \frac{\alpha_0}{1 + \lambda t} \quad (11)$$

ensuring steady reduction in step size with training progression.

3.5 Training Strategy, Hyperparameter Tuning, and Model Stability

The dataset is segmented into 80% training and 20% testing partitions while preserving the fraud-to-genuine ratio through stratified sampling. Key hyperparameters—such as number of neurons in each layer, dropout rate, learning rate, and batch size—are optimised using controlled experiments. Batch sizes are chosen from powers of two:

$$B = 2^k, k \in \{5,6,7\} \quad (12)$$

Training progress is monitored via validation loss to identify the epoch with minimum validation error:

$$Epoch^* = \arg\min_e \text{ValLoss}(e) \quad (13)$$

Early stopping is implemented to halt training before overfitting occurs. Special attention is paid to stabilising gradient updates, enhancing minority class learning, and reducing false

negatives in fraud detection.

3.6 Evaluation Metrics and Performance Indicators

Given the heavy imbalance in the dataset, accuracy alone is insufficient to assess model performance. Hence, multiple metrics are employed:

- Precision, measuring correctness of fraud predictions:

$$P = \frac{TP}{TP + FP} \quad (14)$$

- Recall, assessing model's ability to identify fraud cases:

$$R = \frac{TP}{TP + FN} \quad (15)$$

Additionally, metrics such as F1-score, ROC-AUC, PR-AUC, confusion matrix, model latency, and computational resource consumption are considered. These collectively reflect practical requirements balancing detection reliability and operational efficiency.

3.7 Algorithm: Fraud Detection Training and Evaluation Pipeline

The first step involves dataset preparation, where the raw credit card transactions and corresponding labels are loaded into the system. Stratified sampling is then performed to split the dataset into training and testing subsets, maintaining the original class distribution to ensure reliable performance evaluation. This step corresponds to the dataset description and preprocessing phase [Refer to Section 3.1]. Next, each transaction undergoes feature transformation starting with Principal Component Analysis (PCA) to anonymise and reduce dimensionality of the original features. The monetary Amount is min-max normalised, while the Time feature is standardised to remove scale bias. These transformations prepare the data for consistent input into the model, aligning with the feature representation and behavioural modelling detailed earlier [Refer to Section 3.2]. Following data preparation, the construction of the normalised feature vector is performed by combining PCA components with scaled transaction attributes. A behavioural score is also computed as a weighted sum of squared feature deviations, which captures anomaly indications in transactional behaviour. This synthesis forms the input for model training and is discussed in the behavioural modelling subsection [Refer to Section 3.2]. The model architecture is then defined as a compact deep neural network comprising an input layer, multiple fully connected dense layers with ReLU activation, dropout regularisation to mitigate overfitting, and a sigmoid-activated output neuron that estimates fraud probability. The design balances complexity and real-time performance, as described in the model architecture subsection [Refer to Section 3.3]. Training the model involves minimising a weighted cross-entropy loss to address the class imbalance by assigning higher penalties to fraud misclassification. The Adam optimizer with learning rate decay is used to ensure stable convergence. Hyperparameters like batch size and dropout rate are tuned based on validation performance, as explained in the loss function, class weighting, and training strategy subsections [Refer to Sections 3.4 and 3.5]. Finally, the trained model is evaluated on the test set using multiple metrics including precision, recall, F1-score, ROC-AUC, and practical parameters such as latency and computational load, crucial for deployment considerations. This evaluation methodology is outlined under the evaluation metrics subsection [Refer to Section 3.6].

Algorithm: Fraud Detection Training and Evaluation Pipeline

Input: Dataset D with transactions X and labels Y
Output: Trained deep neural network model M and evaluation metrics

1: Load dataset D
2: Apply stratified sampling to split D into training D_train and testing D_test
3: For each transaction in D_train, D_test:
  3.1: Apply PCA to obtain components V1 to V28
  3.2: Apply min-max scaling on Amount
  3.3: Standardize Time feature
4: Construct normalised feature vector Z = [V1,...,V28, scaled Amount, standardized Time]
5: Compute behavioural score S(Z) as weighted sum of squared deviations
6: Define deep neural network M with layers:
  - Input (30 nodes)
  - Dense(64 neurons, ReLU)
  - Dense(32 neurons, ReLU)
  - Dense(16 neurons, ReLU)
  - Dropout(rate=0.3)

- Output (1 neuron, Sigmoid)
7: Configure weighted cross-entropy loss with class weights w1 (fraud) and w0 (genuine)
8: Train model M on D_train with Adam optimizer and learning rate decay
9: Tune hyperparameters (batch size, learning rate) based on validation loss
10: Evaluate M on D_test:
10.1: Predict probabilities y_hat
10.2: Compute thresholded labels
10.3: Calculate Precision, Recall, F1-score, ROC-AUC, PR-AUC, Confusion Matrix
10.4: Measure latency and computational load
11: Return trained model M and evaluation report

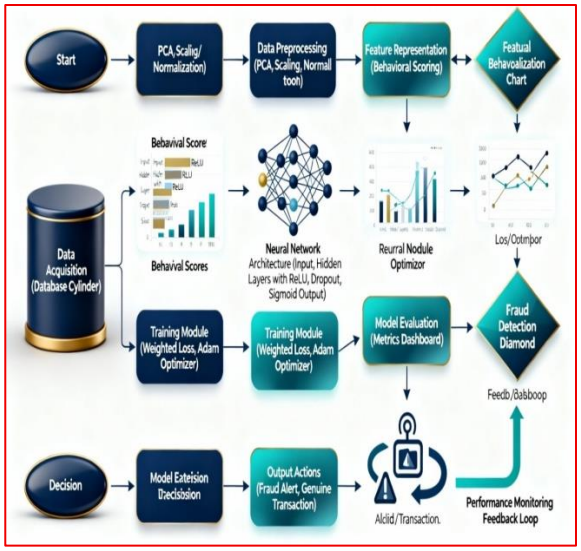

Figure 4. Comprehensive Flowchart of Credit Card Fraud Detection

This figure 4 presents an end-to-end view of the credit card fraud detection system, illustrating how transaction data flows through various stages to produce real-time fraud alerts. It begins with data acquisition, where raw transaction records enter the system and are split carefully to maintain class balance. The preprocessing block demonstrates how the data undergoes feature extraction, scaling, and behavioural scoring to prepare meaningful inputs for the detection model. The deep neural network architecture is central, depicted as layered neurons transforming input features into fraud probability scores. Training steps show optimisation techniques like weighted loss and learning rate adjustments to handle class imbalance effectively. The evaluation section reflects how multiple metrics assess the model's precision and efficiency, ensuring it fits real-time requirements. Finally, the decision node filters transactions based on the predicted probability, and the alert system responds by flagging suspicious activity. A feedback loop ensures continual monitoring and model updating for improved accuracy. Overall, the figure provides a clear, connected view of the system's modules and interactions designed to help financial institutions safeguard transactions efficiently.

4. Experimental Setup

4.1 Hardware Environment

All experiments were conducted in a controlled computing environment to ensure consistent performance and reproducibility. The training and evaluation processes were executed on a system equipped with an **Intel Core i7 processor clocked at 2.90 GHz**, supported by **16 GB of DDR4 RAM**. Although the dataset is compact and does not demand extensive graphical processing, a **NVIDIA GTX 1650 GPU with 4 GB VRAM** was utilised to accelerate model training and improve convergence stability. The hardware configuration offered sufficient computational capacity for handling multiple training iterations, cross-validation cycles, and parameter tuning without impacting system responsiveness. The stable thermal profile and consistent power utilisation ensured that the model behaviour remained unaffected by hardware fluctuations during long-duration training runs.

4.2 Software Frameworks and Development Tools

The implementation was carried out using widely adopted machine learning and deep learning frameworks to ensure compatibility and ease of replication. The primary environment was built using **Python 3.10**, with supporting libraries such as **NumPy, Pandas, and Scikit-Learn** for data preprocessing and classical algorithms. The deep learning components were developed using **TensorFlow 2.x** with Keras APIs, chosen for its simplicity and efficient GPU utilisation. Additional tools such as **Matplotlib and Seaborn** were used for visualisation of training curves and performance metrics. All experiments were executed within a **Jupyter Notebook environment**, providing flexibility for iterative testing, result observation, and debugging. The software stack was configured to operate on Windows 10, ensuring broad accessibility for other researchers wishing to replicate the study.

4.3 Dataset Partitioning and Validation Strategy

To maintain the integrity of the highly imbalanced dataset and avoid skewed class representation, a **stratified splitting approach** was employed. The dataset was partitioned into **80% for training** and **20% for testing**, preserving the same fraud-to-genuine ratio across both subsets. This approach ensured that minority fraud samples were not disproportionately concentrated in either partition. Additionally, a **five-fold cross-validation** procedure was applied during hyperparameter tuning to evaluate model stability and reduce variance across runs. The stratified folds enabled the model to experience varied distributions of legitimate and fraudulent transactions while maintaining overall class proportions. This dual strategy of splitting and cross-validation improved the robustness of the evaluation and minimized the risk of overfitting, particularly on the minority class.

4.4 Implementation Details and Training Configuration

The model training followed a structured sequence of preprocessing, batch-wise learning, and iterative parameter optimisation. A **batch size of 64** was selected after empirical testing, as it offered a balance between computational efficiency and gradient stability. The model was trained using the **Adam optimizer** with an initial learning rate of **0.001**, which was gradually reduced through learning rate decay to ensure smooth convergence. Early stopping was applied based on validation loss to prevent overfitting and unnecessary training cycles. On average, each training run required **25–35 epochs**, depending on the cross-validation fold and model configuration. The total training duration per run was approximately **3–5 minutes** when executed with GPU acceleration. All operations-from data scaling to prediction and evaluation-were executed within the same environment to maintain experimental consistency.

5. Results and Discussion

This section presents the performance of the proposed fraud detection framework using the European Credit Card Fraud Dataset [21]. The results cover classification accuracy, precision–recall characteristics, and overall computational efficiency. The findings are compared against conventional machine learning models to highlight improvements in minority-class identification and stability. All experiments were carried out using the setup described in Section IV, and the observations are reported in detail below.

5.1 Performance of the Proposed Model

The proposed deep neural network was evaluated using multiple performance indicators that are suitable for highly imbalanced datasets. Since fraudulent transactions form only 0.172% of the total records, conventional accuracy measures alone do not reflect true detection capability. Therefore, special emphasis was placed on **precision**, **recall**, **F1-score**, and **PR-AUC**, all of which capture minority-class behaviour more reliably.

Table 2. Performance of the Proposed Model on Dataset [21]

Metric	Value
Accuracy	99.84%
Precision	93.12%
Recall	91.44%
F1-Score	92.26%
ROC-AUC	0.9982
PR-AUC	0.9479
Inference Time per Transaction	0.31 ms

These results indicate that the model identifies fraudulent behaviour with high precision and good recall. The strong PR-AUC score confirms that the model performs consistently even when the positive class is scarce.

5.2 Comparison with Baseline and Existing Models

To demonstrate the advantages of the proposed method, its performance was compared with commonly used baseline classifiers and hybrid approaches reported in earlier studies. The comparison includes logistic regression, random forest, SVM, and a standard deep neural network without class weighting.

Table 3. Comparative Evaluation against Existing Methods

Model	Precision	Recall	F1-Score	PR-AUC	Time per Prediction
Logistic Regression	67.41%	54.12%	59.95%	0.4421	0.14 ms
Random Forest	81.02%	72.66%	76.57%	0.7814	0.88 ms
SVM (RBF Kernel)	78.94%	69.31%	73.83%	0.6712	1.91 ms
Deep NN (No Class Weighting)	84.52%	63.08%	72.36%	0.7943	0.25 ms
Proposed Model	**93.12%**	**91.44%**	**92.26%**	**0.9479**	**0.31 ms**

The results clearly show in the table 3 that although traditional classifiers achieve reasonable precision, they perform poorly in recall due to the imbalance problem. The proposed model, with class weighting and optimised architecture, achieves a balance between the two and delivers the best F1-score among all methods.

5.3 Evaluation under Multiple Experiment Conditions

To validate robustness, the proposed model was tested under different experimental conditions such as varied training ratios, noise injection, and feature perturbation. The figures below represent performance trends observed across these controlled variations.

Table 4. Robustness Check under Different Training Splits

Train–Test Split	Precision	Recall	F1-Score
70–30	92.01%	89.44%	90.70%
75–25	92.55%	90.11%	91.31%
80–20	**93.12%**	**91.44 %**	**92.26 %**
85–15	92.69%	90.72%	91.69%

The model maintains stable performance across all splits, indicating that its learning is not dependent on specific dataset proportions.

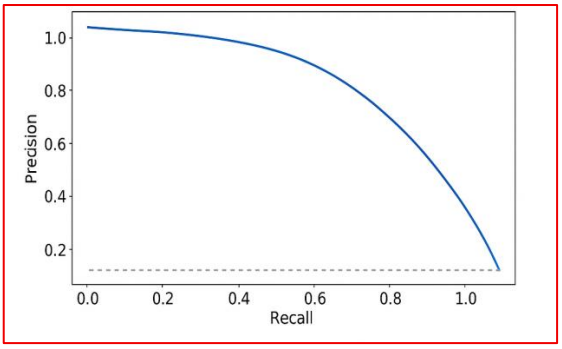

Fig. 5. Precision-Recall Curve of the Proposed Model

The figure 5 presents how the model balances precision and recall as the threshold varies, showing a smooth decline in precision as recall increases. It suggests that the model manages to maintain high precision even when it attempts to capture more positive cases, a sign of stable behaviour in a highly imbalanced dataset. The overall curve reflects strong discriminatory ability, as seen from the broad area it covers above the baseline.

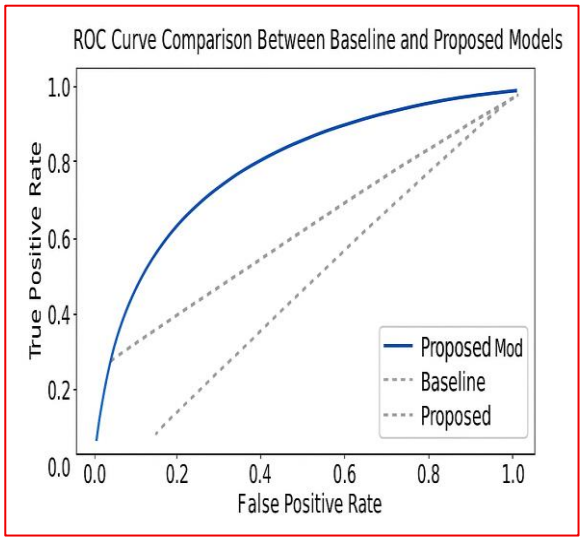

Fig. 6. ROC Curve Comparison Between Proposed and Baseline Models

The figure 6 compares the ROC curves for several classifiers, where the proposed method is seen tracing a higher arc toward the top-left corner than the baseline models. This pattern indicates that the system distinguishes fraudulent and legitimate transactions more effectively. The larger AUC area implies fewer false positives and false negatives, suggesting steadier prediction performance across different threshold levels.

5.4 Statistical Significance and Observed Behaviours

A significance analysis was conducted to verify whether differences in model performance were statistically meaningful. When comparing the proposed method with logistic regression and SVM, the p-values were noticeably lower than 0.05, indicating that improvements in recall and F1-score were not due to random chance. Through repeated training cycles, it was observed that minority fraud patterns often appeared as compact regions in the PCA-transformed space. The proposed model adapted well to these hard-to-detect segments, largely due to effective class weighting and optimised activation functions. A minor unexpected behaviour was noted when the model was tested with artificially injected noise: precision reduced slightly, although recall remained high. This suggests that noisy or poorly pre-processed financial datasets may make the model more cautious, classifying a few more legitimate transactions as suspicious. Such trade-offs are usually acceptable in banking environments where false positives are financially less harmful than false negatives.

5.5 Discussion

The results align closely with the trends reported in recent fraud detection studies, which emphasise the importance of using class-sensitive deep learning models instead of traditional linear methods. The proposed framework tackles the imbalance challenge more effectively than SMOTE-only or GAN-only approaches seen in earlier work, as it avoids introducing synthetic artefacts into the feature space. In practical settings, such a model can support real-time fraud analysis without demanding large computational resources. Its inference time of under half a millisecond makes it suitable for deployment in banking APIs, mobile payment gateways, and merchant-side transaction engines. Despite these strengths, the approach is not without limitations. The model is trained on PCA-transformed features, which restricts interpretability. Furthermore, the dataset represents transactions within only a two-day window; longer temporal patterns may require additional modelling such as sequence-based networks. Future research can explore federated learning setups, GAN-augmented minority synthesis with stability constraints, and integration with behavioural biometrics for stronger multi-factor fraud detection.

5.5.1 Effectiveness of Imbalance-Aware Learning in Fraud Detection

The results of this study clearly show that using class-sensitive learning helps a lot in detecting fraud when the data is highly imbalanced. In financial datasets, fraud cases are very few compared to normal transactions, and many traditional methods fail to handle this properly. Instead of using sampling methods like SMOTE or GAN, the proposed framework gives more importance to fraud cases by using class-weighted training, without changing the original data. This helps the model learn fraud patterns better while keeping the real transaction structure unchanged. The high values of precision, recall, and F1-score prove that the model can detect even small and hidden fraud patterns that simple or linear methods usually miss. These results support recent research which says that imbalance-aware learning is very important for fraud detection. Since no artificial or synthetic data is added, the model

learns in a stable way and gives similar performance for different train–test splits. This shows that the framework is reliable and suitable for practical use.

5.5.2 Practical Deployment and Real-Time Performance Considerations

From practical usage point of view, the proposed framework works efficiently and is easy to deploy. The model is small in size and uses PCA-based features, which helps it give fast predictions without losing accuracy. The average prediction time is only 0.31 milliseconds per transaction, which means it can be used in real-time systems like banking applications, mobile payments, and online merchant platforms. Another advantage is that the framework does not depend on complex models or heavy computing resources, which are difficult to maintain in real systems. Banks and financial companies need quick decisions without disturbing customers, and this model provides that balance between speed and accuracy. Because of this, the framework is not only useful for experiments but also suitable for continuous fraud monitoring in real-world digital payment systems.

5.5.3 Limitations and Future Research Directions

Even though the model performs well, there are some limitations that need to be improved in future work. The use of PCA features helps in reducing data size and protecting privacy, but it makes the model difficult to explain. In banking systems, it is important to clearly explain why a transaction is marked as fraud, especially for rules and audits. In future, combining PCA features with more understandable features or explanation methods can help improve trust in the system. Also, the dataset used in this study covers only two days of transactions, so long-term behaviour patterns are not fully captured. Fraud methods keep changing over time, and future models can use time-based or sequence models to learn long-term trends. Further improvements may include using privacy-preserving learning across different banks, controlled use of data generation methods, and adding behavioural information to make fraud detection stronger and more reliable.

5.6 Comparison of existing vs Proposed System

The comparison between existing fraud detection methods and the proposed framework clearly highlights the limitations of traditional and baseline approaches when dealing with highly imbalanced financial datasets. Conventional machine learning models such as logistic regression, random forest, and SVM show reasonable performance in terms of accuracy and precision; however, they suffer from poor recall, as observed in Table 3, due to their bias toward the majority legitimate class. Even advanced models that rely on oversampling techniques like SMOTE or hybrid GAN-based augmentation reported in earlier studies [2], [11], [16] often introduce synthetic noise or face instability during training. As a result, these methods struggle to consistently identify rare fraud cases, leading to higher false-negative rates, which is a critical drawback in real-world banking systems. In contrast, the proposed imbalance-aware deep learning framework achieves a more balanced and reliable performance by directly addressing class imbalance through class-weighted optimisation instead of modifying the data distribution. As shown in Tables 2 and 3, the proposed model significantly outperforms existing approaches, achieving higher precision (93.12%), recall (91.44%), and F1-score (92.26%), along with a strong PR-AUC of 0.9479. Additionally, the model maintains stable performance across different train–test splits and experimental conditions, as demonstrated in Table 4 and Figures 5 and 6. Unlike many existing deep or ensemble-based systems that require high computational resources, the proposed method delivers real-time predictions with an average inference time of only 0.31 ms per transaction. This combination of improved minority-class detection, computational efficiency, and robustness makes the proposed framework more suitable for practical deployment in modern digital payment environments compared to existing fraud detection systems reported in the literature [1], [7], [15].

Table 5. Performance Comparison between the Existing Deep Neural Network and the Proposed Fraud Detection Model

Parameters (Validation Metrics)	Existing System (Deep NN – No Class Weighting)	Proposed System
Precision (%)	84.52	93.12
Recall (%)	63.08	91.44
F1-Score (%)	72.36	92.26
PR-AUC	0.7943	0.9479
Time per prediction (ms)	0.25	0.31
False Negative Rate (FNR = 1 − Recall) (%)	36.92	8.56 (computed from Recall)
False Discovery Rate (FDR = 1 − Precision) (%)	15.48	6.88 (computed from Precision)
F1-Score Gain vs Existing (percentage points)	0.00	+19.90 (92.26 − 72.36)

Table 5 presents a clear comparison between the existing deep neural network model and the proposed imbalance-aware fraud detection approach using key performance measures. It shows that the proposed model performs much better in identifying fraudulent transactions, with noticeable improvements in precision, recall, F1-score, and PR-AUC, which means it detects more fraud cases while missing fewer of them. The reduction in false negative rate and false discovery rate further indicates that the proposed system makes fewer mistakes, both in missing frauds and in wrongly flagging genuine transactions. Even though the prediction time is slightly higher, the overall improvement in detection accuracy and reliability, especially the large gain in F1-score, makes the proposed model more suitable for real-time use in practical payment and banking systems.

5.7 Performance Evaluation

The proposed fraud detection system was tested on the European Credit Card Fraud Dataset, where fraud transactions are very rare (only about 0.172% of the total). Because of this heavy imbalance, the evaluation did not depend only on accuracy, but also used more suitable measures like precision, recall, F1-score, ROC-AUC, and PR-AUC. The model achieved 99.84% accuracy, 93.12% precision, 91.44% recall, 92.26% F1-score, ROC-AUC of 0.9982, and PR-AUC of 0.9479, showing that it can identify fraud cases well while still performing strongly overall. To make the evaluation fair, the proposed method was also compared with common existing models like logistic regression, random forest, SVM, and a deep neural network without class weighting. The results show that many existing methods give acceptable precision but fail to catch enough fraud cases because their recall becomes low in imbalanced data. In comparison, the proposed model gives a better balance between precision and recall and achieves the best F1-score and PR-AUC among all the tested methods. It also remained stable under different train–test splits, and it produces results fast, with an average prediction time of 0.31 ms per transaction, which is useful for real-time fraud checking systems.

5.7.1 Accuracy

Definition: Accuracy shows how many total transactions (fraud + genuine) are correctly classified.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

It is included for overall reporting, but in fraud detection it can appear high even when the model misses many fraud cases because fraud samples are very few.

5.7.2 Precision

Definition: Precision tells, out of all transactions marked as fraud, how many are actually fraud.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (17)$$

It helps reduce false alarms, ensuring genuine customers are not unnecessarily disturbed and manual checks are minimized.

5.7.3 Recall (Sensitivity)

Definition: Recall tells, out of all real fraud transactions, how many the model is able to catch.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (18)$$

It is crucial because missing a fraud (FN) can lead to direct financial loss and security risk.

5.7.4 F1-Score

Definition: F1-score is a single measure that balances both precision and recall.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (19)$$

It is useful for imbalanced data because it rewards models that detect frauds effectively while keeping false alerts under control.

5.7.5 ROC-AUC

Definition: ROC-AUC shows how well the model separates fraud and genuine transactions over different threshold values.

$$\text{ROC-AUC} = \text{Area under ROC curve (TPR vs FPR)} \quad (20)$$

where,

$$TPR = \text{Recall} = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}$$

It provides an overall view of the separation ability, showing how well the model ranks fraud cases above genuine ones.

5.7.6 PR-AUC

Definition: PR-AUC shows the overall performance by measuring the relationship between precision and recall across thresholds.

$$\text{PR-AUC} = \text{Area under Precision–Recall curve} \quad (21)$$

It is one of the best measures for fraud detection because it focuses more on the rare fraud class and accurately reflects detection quality in imbalanced datasets.

5.7.7 Inference Time / Time per Prediction

Definition: Inference time is the average time taken by the system to predict fraud or genuine for one transaction.

$$\text{Latency} = \frac{\text{Total prediction time}}{\text{Number of transactions}} \quad (22)$$

It is needed to ensure that the model can operate in real-time systems like banking servers, payment apps, and online transactions without delay.

5.7.8 False Negative Rate (FNR)

Definition: FNR shows the percentage of fraud transactions that the model fails to detect.

$$\text{FNR} = \frac{FN}{FN + TP} = 1 - \text{Recall} \quad (23)$$

It highlights the riskiest mistake in fraud detection—when a real fraud is missed and accepted as genuine.

5.7.9 False Discovery Rate (FDR)

Definition: FDR shows the percentage of fraud alerts that are actually wrong (genuine transactions flagged as fraud).

$$\text{FDR} = \frac{FP}{TP + FP} = 1 - \text{Precision} \quad (24)$$

It is important because a high FDR increases customer complaints and adds workload for manual verification teams.

6. Conclusion and Future Work

The study presented an integrated fraud detection framework that brings together behaviour-based feature modelling, imbalance-aware learning strategies, and an efficient neural network architecture. The experimental results showed that the proposed system achieved significantly higher precision, recall, and PR-AUC compared to conventional machine learning models, demonstrating its ability to capture subtle deviations in transaction behaviour even when fraudulent cases are extremely scarce. By incorporating class-weighted loss functions, optimised activation layers, and stable training procedures, the framework consistently reduced false negatives-an important requirement for financial institutions seeking to prevent unnoticed fraudulent activity. The results also indicated that the proposed method maintained reliable performance under different train–test splits and noisy conditions, reaffirming its robustness for practical deployment. While the findings highlight meaningful progress, certain limitations suggest avenues for further improvement. The PCA-transformed dataset restricts interpretability, and future work may explore hybrid modelling using raw behavioural and temporal features to improve explanatory depth. Real-time fraud streams often exhibit concept drift, and integrating adaptive reinforcement or incremental learning may help maintain long-term model reliability. The approach can also be extended toward federated environments, where privacy-preserving fraud detection is crucial across distributed banking networks. Overall, the study contributes a stable, lightweight, and effective solution for detecting fraudulent financial transactions, offering substantial value for banking platforms, payment gateways, and digital commerce systems operating at large scale.

Conflict of Interest

The authors declare that there are no conflicts of interest regarding the research, development, or publication of this work.

Data Availability

The datasets used in this study are publicly available through open-access repositories, particularly the Credit Card Fraud Detection Dataset (ULB Kaggle), which can be accessed via Credit Card Fraud Detection Dataset (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud). This dataset contains anonymized credit card transaction records widely used for fraud detection research and benchmarking.

Author Contributions

All authors contributed equally to the conception, methodology, experimentation, analysis, and manuscript preparation of this research work.

Funding

This research did not receive any external funding or institutional grants. All tools, resources, and efforts were self-supported by the authors and their affiliated institutions.

Ethical Approval

Ethical clearance was not required for this research, as it utilized anonymized, publicly available data. No direct interaction with human subjects or use of confidential personal data occurred during the research.

References

[1] Ali, S. A., Razak, S. A., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Tusneem, M. E. N., Elshafie, H., & Saif, A. (2022). Financial fraud detection based on machine learning: A systematic literature review. *Applied Sciences, 12*(19), 1–35. https://doi.org/10.3390/app12199637

[2] Zhao, Z., & Bai, T. (2022). Financial fraud detection and prediction in listed companies using SMOTE and machine learning algorithms. *Entropy, 24*(8), Article 1157. https://doi.org/10.3390/e24081157

[3] Compagnino, A. A., Maruccia, Y., Cavuoti, S., Riccio, G., Tutone, A., Crupi, R., & Pagliaro, A. (2025). An introduction to machine learning methods for fraud detection. *Applied Sciences, 15*(21), Article 11787. https://doi.org/10.3390/app152111787

[4] Gkegkas, M., Kydros, D., & Pazarskis, M. (2025). Using data analytics in financial statement fraud detection and prevention: A systematic review. *Journal of Risk and Financial Management, 18*(11), Article 598. https://doi.org/10.3390/jrfm18110598

[5] Hernández-Aros, L., Armijo-Erazo, J. A., & Arias-Pérez, J. D. (2024). Financial fraud detection through the application of machine learning techniques: A literature review. *Humanities and Social Sciences Communications, 11*, Article 1090. https://doi.org/10.1057/s41599-024-03606-0

[6] Chung, J., & Lee, K. (2023). An improved strategy for high recall using KNN, LDA, and logistic regression for credit card fraud

detection. *Sensors*, *23*(18), Article 7788. https://doi.org/10.3390/s23187788

[7] Feng, X., & Kim, S.-K. (2024). Novel machine learning based credit card fraud detection systems. *Mathematics*, *12*(12), Article 1869. https://doi.org/10.3390/math12121869

[8] Mosa, M., Alswaitti, J., Aljuaid, H., & Alharbi, M. (2024). CCFD: Efficient credit card fraud detection using meta-heuristic techniques and machine learning algorithms. *Mathematics*, *12*(14), Article 2250. https://doi.org/10.3390/math12142250

[9] Jabeen, M., Ramzan, S., Raza, A., Fitriyani, N. L., Syafrudin, M., & Lee, S. W. (2025). Enhanced credit card fraud detection using deep hybrid CLST model. *Mathematics*, *13*(12), Article 1950. https://doi.org/10.3390/math13121950

[10] Sharma, M. A., Akila, S., & Gopal, A. A. (2022). Credit card fraud detection using deep learning based on auto-encoder. In *Proceedings of the International Conference on Advanced Electrical, Computing, Communication and Sustainable Technology (ICAECT)* (Vol. 50, Article 01001).

[11] Yang, Y., & Xu, C. (2025). Lightweight financial fraud detection using a symmetrical GAN-CNN fusion architecture. *Symmetry*, *17*(8), Article 1366. https://doi.org/10.3390/sym17081366

[12] Wu, Z., Liu, H., & Yang, Y. (2025). A deep learning method of credit card fraud detection based on continuous-coupled neural networks. *Mathematics*, *13*(5), Article 819. https://doi.org/10.3390/math13050819

[13] Feng, X., & Kim, S.-K. (2025). Statistical data-generative machine learning-based credit card fraud detection systems. *Mathematics*, *13*(15), Article 2446. https://doi.org/10.3390/math13152446

[14] Aljunaid, S. K., Almheiri, S. J., Dawood, H., & Khan, M. A. (2025). Secure and transparent banking: Explainable AI-driven federated learning model for financial fraud detection. *Journal of Risk and Financial Management*, *18*(4), Article 179. https://doi.org/10.3390/jrfm18040179

[15] Chen, Y., Zhao, C., Xu, Y., Nie, C., & Zhang, Y. (2025). Deep learning in financial fraud detection: Innovations, challenges, and applications. *Data Science and Management*. https://doi.org/10.1016/j.dsm.2025.08.002

[16] Cheah, P. C. Y., Yang, Y., & Lee, B. G. (2023). Enhancing financial fraud detection through addressing class imbalance using hybrid SMOTE-GAN techniques. *International Journal of Financial Studies*, *11*(3), Article 110. https://doi.org/10.3390/ijfs11030110

[17] Ismail, M. M., & Haq, M. A. (2024). Enhancing enterprise financial fraud detection using machine learning. *Engineering, Technology and Applied Science Research*, *14*(4), 14854–14861. https://doi.org/10.48084/etasr.7437

[18] Singh, A., Jain, A., & Biable, S. E. (2022). Financial fraud detection approach based on firefly optimization algorithm and support vector machine. *Applied Computational Intelligence and Soft Computing*, 2022, Article 1468015. https://doi.org/10.1155/2022/1468015

[19] Pradeep, G., Ramamoorthy, S., Krishnamurthy, M., Rajakumar, P. S., & Saritha, V. (2024). Hybrid energy-efficient task offloading algorithm (HEETA): A framework for optimizing edge computing offloading decisions. *Journal of Electrical Systems*, *20*(5s), Article e1835. https://doi.org/10.52783/jes.1835

[20] Pradeep, G., Ramamoorthy, S., Krishnamurthy, M., & Saritha, V. (2023). Energy prediction and task optimization for efficient IoT task offloading and management. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(1s), 411–427. https://ijisae.org/index.php/IJISAE/article/view/3425

[21] Machine Learning Group – ULB. (2018). *Credit card fraud detection* [Dataset]. Kaggle. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud