Journal of Applied Data Sciences
Vol. 6, No. 4, December 2025, pp. 2838 -2857

An Explainable Credit Card Fraud Detection Model using Machine Learning and Deep Learning Approaches

Mona Alkhozae[1,*], Miada Almasre[2], Abeer Almakky[3], Reemah M. Alhebshi[4], Amani Alamri[5], Widad hakami[6], Lamia Alshahrani[7]

[1,2,3,5,6,7]*Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

[4]*Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

(Received: April 01, 2025; Revised: June 24, 2025; Accepted: September 22, 2025; Available online: October 04, 2025)

Abstract

This study proposes an adaptive, interpretable real-time fraud detection and prevention system designed for high-risk financial environments, capable of processing over 1.6 million imbalanced credit card transactions with low latency. The objective is to build a unified framework that integrates predictive accuracy, explainability, and adaptability. The methodology follows four phases: exploratory data analysis to reveal structural and behavioral fraud patterns, feature engineering with domain-informed attributes and ADASYN oversampling to mitigate the 1:174 imbalance, training of multiple models (XGBoost, LightGBM, Random Forest, Gradient Boosting, and MLP), and an ensemble architecture evaluated with SHAP-based explainability. The system introduces three key contributions: stability-aware SHAP caching that reduces explanation latency to 41.2 ms, reinforcement learning–based threshold tuning that dynamically adapts to evolving fraud patterns, and out-of-distribution detection to enhance resilience against data drift. Results demonstrate strong performance, with XGBoost achieving 99.86% accuracy, 96.36% precision, 80.59% recall, F1-score of 0.878, and ROC-AUC of 0.9988, outperforming other models. The full system attained 93.2% accuracy, 90.2% F1-score, and 96.1% AUC at the system level, successfully blocking 91% of fraudulent transactions while maintaining a false positive rate of 7.8%. Novelty lies in combining explainability and adaptivity in a production-ready architecture, where reinforcement learning enables continuous threshold self-regulation and SHAP stability analysis validates interpretability across models. These findings show that high fraud detection accuracy and transparency are not mutually exclusive, offering a scalable blueprint for financial institutions and other critical domains requiring real-time, explainable, and adaptive decision-making.

Keywords: Credit Card, Fraud Detection, Machine Learning, Deep Learning, Large Language Models (LLM), SHAP Values, Neural Network

1. Introduction

In today's interconnected digital economy, credit card processing underpins a vast majority of commercial transactions, chiefly because of the efficiency and accessibility it affords. Yet, as digital infrastructure expands, so too do the vectors for exploitation. The sheer velocity of digitization has been paralleled by a troubling rise in fraudulent activity, now recognized as a formidable threat not only to individual consumers but also to institutional and financial stakeholders at large. As reported in [1], financial damages attributed to credit card fraud reached an estimated $32.39 billion globally in 2020 and are projected to surpass $40 billion by 2027. Historically, many institutions have relied on rule-based systems to combat fraud. However, these conventional approaches have shown a persistent lag behind the increasingly inventive and adaptive strategies employed by fraudsters. The result is often sluggish detection and inflated false positive rates, culminating in higher operational costs and erosion of customer trust [1]. In response to this shortfall, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as more robust alternatives, capable of discerning nuanced behavioral signals embedded within complex transaction datasets [2], [3]. Simultaneously, progress in Explainable Artificial Intelligence (XAI) has addressed growing concerns about the opacity of such data-driven models. Tools like LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) have proven especially useful in critical, high-risk applications such as financial fraud

***Corresponding author: Mona Alkhozae (malkhozae@kau.edu.sa)**
DOI: https://doi.org/10.47738/jads.v6i4.962

ISSN 2723-6471

prevention [2]. Their utility lies not only in offering clarity and interpretability to users and institutions but also in enhancing the underlying model's accountability and acceptance [1], [4].

Technical progress has not resolved all obstacles. A persistent challenge is the skewed class distribution inherent in fraud datasets, which tends to distort learning dynamics and weaken predictive accuracy. Compounding this, many advanced ML models continue to operate as "black boxes," rendering them unsuitable for use in strictly regulated financial ecosystems. While solutions such as oversampling, ensemble methods, and interpretability frameworks like SHAP have been proposed, they frequently remain static in their behavior and fail to adapt in real time to the dynamic nature of fraudulent activities. Moreover, despite the interpretability gains offered by SHAP, practical deployment often runs into performance bottlenecks. The computational burden associated with SHAP, particularly in production pipelines, can be considerable. Most real-world systems also lack provisions for dynamic recalibration; for instance, they do not adapt threshold decisions in response to fluctuating false positive or false negative patterns. This leads to an unfortunate trade-off in system design: one must often choose between speed, interpretability, and adaptability. Typically, systems end up excelling in only one of these dimensions, either performant but opaque, interpretable but slow, or adaptive yet poorly tuned.

Overcoming these multi-faceted challenges requires a robust framework that integrates interpretability, real-time dynamic adaptability, and does not neglect efficiency or reliability. Therefore, this paper aims to present a real-time fraud detection and prevention mechanism that utilizes ML and DL techniques enhanced with SHAP interpretability. Based on the previous aim, our research objectives are twofold. First, we focus on developing machine learning and deep learning models for credit card fraud detection, ensuring that these models incorporate SHAP-based interpretability and domain-informed feature engineering. Second, we aim to assess the performance of the developed models using established evaluation metrics such as accuracy, recall, F1-score, and AUC-ROC, thereby providing a comprehensive understanding of their effectiveness.

The contributions of this paper can be outlined as follows: 1) a unified, four-phase method that addresses real-time fraud detection, models' explainability, and fraud prevention, a mechanism that is designed for decision-making contexts. 2) The pipeline presents a stability-aware explanation approach that verifies SHAP consistency across multiple inference cycles. 3) A dynamic threshold tuning component is also used to adjust decision boundaries in response to shifting misclassification patterns. And 4) to ensure feasibility in real-world contexts, SHAP caching and latency benchmarking techniques are used to reduce overhead and optimize runtime responsiveness. Our validation pipeline includes ablation studies, ensemble agreement diagnostics, and robustness assessments under Out-Of-Distribution (OOD) conditions, as these evaluations provide a comprehensive view of system performance in both expected and novel operational scenarios. Hence, our paper will discuss the following: Section 2 offers an analysis of related literature, while Section 3 discusses the methods used for fraud detection and prevention in our project. Section 4 presents our findings in the context of research objectives, and Section 5 is a conclusion that outlines future research directions.

2. Related Work

This section reviews recent advances in credit card fraud detection, focusing on machine learning, deep learning, and interpretability techniques, while noting gaps in real-time prevention and class imbalance management. Mill et al. [1] examined the use of XAI in real-time fraud detection, emphasizing regulatory drivers like Strong Customer Authentication (SCA) and identifying four research goals: embedding explanations in operational contexts, prioritizing intrinsically interpretable models, quantitatively evaluating explanation–model alignment, and tailoring explanations to stakeholders. They stressed the need for human-centered explainability, real-world validation, and greater trust before adoption by financial institutions. In [2], the authors integrated XAI with ML for fraud detection, using SHAP and LIME for interpretability, SMOTE for class balancing, and XGBoost for robust performance, achieving 96.64% accuracy, 94.79% AUC-ROC, and 92.92% recall. Aljunaid et al. [3] proposed an Explainable Federated Learning (XFL) model that combines privacy-preserving federated training with SHAP and LIME explanations, using a DNN with Auto-Encoders and RBMs to capture complex fraud patterns. Tested on real financial data, it achieved 99.95% accuracy, 99.95% sensitivity, 100% PPV, and a 0.05% miss rate, demonstrating high performance, interpretability, and regulatory compliance. Mallam et al. [4] evaluated supervised learning models, including logistic regression, kNN,

SGD, SVM, Extra Trees, Random Forest, and MLP on an imbalanced credit card fraud dataset. Random Forest performed best with 98.5% accuracy and 0.99 AUC, though the study noted overfitting and class imbalance as limitations. Mir [5] proposed an adaptive fraud detection framework using real-time ML with streaming data, combining decision trees, SVM, and ensembles. Adaptive models outperformed static ones, especially for complex fraud patterns, but faced challenges in feature selection and computational cost. Patel [6] reviewed credit risk and fraud detection methods using big data analytics, highlighting the shift from rule-based to advanced ML/DL approaches, and addressing issues of data security, scalability, evolving fraud patterns, regulatory compliance, and the role of AI, data quality, IoT, and blockchain in mitigating privacy and imbalance challenges.

The study in [7] compared multiple ML algorithms for fraud detection, addressing data imbalance and privacy concerns, and proposed a hybrid ANN–federated learning approach for decentralized training. The federated ANN achieved up to 99.96% accuracy while mitigating overfitting and GDPR-related sharing issues, though challenges remained in computation, dataset access, and anomaly detection. Bharath et al. [8] developed a Python-based ANN system using features such as interaction, amount, and intervals, achieving high accuracy, precision, and recall; they noted evolving fraud patterns and imbalance, recommending hybrid models and better feature selection for adaptability. Baisholan and Baqapuri [9] focused on interpretability and imbalance, employing XGBoost and Random Forest with class weighting and threshold adjustments, achieving 97% AUC-PR and 95% recall. SHAP enhanced transparency, and threshold optimization was prioritized over oversampling. Ojo and Tomy [10] examined SHAP and LIME for building trust in fraud detection, achieving 96% accuracy and 0.95 ROC-AUC on the Kaggle dataset using Random Forest and Gradient Boosting with SMOTE, highlighting the regulatory need to balance performance and interpretability. In [11], a hybrid model combining Gradient Boosting, Random Forest, and Feedforward Neural Networks achieved 98.8% accuracy, 80.2% recall, and 0.96 AUC, processing transactions in 500 ms with SMOTE and dimensionality reduction, and emphasizing interpretability and unsupervised learning to address evolving fraud. In [12], a privacy-preserving framework combined Federated Learning with XAI, using a DNN enhanced by Auto-Encoders and RBMs to detect complex fraud patterns in imbalanced datasets. SHAP and LIME enabled traceable, verifiable decisions, and tests on real bank data confirmed high performance, interpretability, and regulatory compliance.

Owoade et al. [13] addressed the growing issue of digital credit card fraud. They proposed a comprehensive framework that combines smart queue systems, machine learning, and regression testing for enhanced fraud detection and response. The model included real-time fraud monitoring, transaction risk classification, and regression testing to ensure the ongoing reliability of the system. Machine learning assessed historical data to adapt to changing fraud strategies, whereas smart waiting systems focused on high-risk transactions, enabling customized intervention. The approach showed improved accuracy, scalability, and response time. However, it faced challenges in some cases due to data protection concerns, high computational requirements, and inadequate human control. They concluded that integrating these technologies gives an adaptable and dynamic solution for modern financial fraud risks. Hasan et al. [14] reviewed the use of XAI and ML in credit card fraud detection, addressing black-box limitations in trust and compliance. They evaluated Random Forest, SVM, Logistic Regression, and ANN, all achieving ~99% accuracy due to dataset imbalance; SVM had the highest recall (89.5%) and ANN the highest precision (79.4%). They recommended inherently interpretable models like decision trees and rule-based systems to improve transparency, meet regulations, and reduce bias. Priya and Sarada [15] reviewed ML applications for fraud detection, noting the shortcomings of reactive methods. They proposed a two-step approach: identifying prior fraud patterns and enhancing authentication via a centralized global fraud database. By testing decision trees, SVM, KNN, Random Forest, and ensembles, they found that hybrid and ensemble models generally outperform and recommended cross-institutional collaboration and AI-driven platforms for scalable, proactive fraud defense.

Habibpur et al. [16] proposed a deep learning–based fraud detection system integrating Uncertainty Quantification (UQ) to improve reliability. Using Monte Carlo Dropout and Ensemble MCD, they assessed prediction confidence via metrics like predictive cross-entropy and reliability plots. Trained on a balanced dataset of 385 features, the ensemble reduced false positives, improved accuracy, and achieved UAcc = 0.85. They recommended hybrid UQ methods and hard leakage detection for better real-time performance. Esenogho et al. [17] combined LSTM networks with AdaBoost and SMOTE-ENN resampling to address class imbalance, leveraging LSTM's ability to capture transaction behavior.

Their method outperformed traditional ML models, achieving 99% AUC, 99.6% sensitivity, and 99.8% specificity, highlighting the challenge posed by the imbalance between legitimate and fraudulent transactions. Gonzalez [18] examined interpretable models for financial fraud detection, comparing SHAP and LIME in high-transparency contexts like accounting and auditing. Using U.S. SEC AAER data (1990–2023) classified as deceptive or non-deceptive, the study trained MLNs, AdaBoost, and XGBoost with Optuna optimization. AdaBoost achieved 97% accuracy, 95.2% weighted accuracy, and 96% recall. SHAP and LIME highlighted key financial metrics, enabling transparent and regulation-compliant detection. Almalki and Masud [19] addressed black-box transparency gaps by combining high-performing ML models with SHAP interpretation on the IEEE-CIS dataset. XGBoost achieved 98.89% accuracy, 93.15% recall, and 0.993 AUC, closely followed by Random Forest with 98.65% accuracy and 0.991 AUC. SHAP identified critical predictors like current asset ratio and expense-to-revenue ratio, showing that robust models combined with explainability improve reliability, auditability, and compliance. All the studies discussed revolve around fraud detection and offer promising prediction algorithms for preventing fraud. Table 1 compares these studies methods, explainability techniques, and performance.

Table 1. Literature Survey

	Methods used	Explainability Techniques	Performance
[1]	A conceptual framework for XAI use in real-time fraud detection: Decision Trees, Risk Scoring Systems (RiskSLIM), Post-hoc models (e.g., SHAP)	Intrinsic interpretability, post-hoc feature attribution (e.g., SHAP, saliency maps)	Not evaluated (conceptual study; no empirical metrics reported)
[2]	XGBoost with SMOTE and Feature Engineering	SHAP, LIME	Accuracy: 96.64%; Recall: 92.92%; AUC-ROC: 94.79%; F1-score: 22.99%
[3]	Deep Neural Network (DNN), Auto-Encoders, Restricted Boltzmann Machines (RBM)	SHAP, LIME	Accuracy: 99.95%, Sensitivity: 99.95%, PPV: 100%, Miss Rate: 0.05%
[4]	Random Forest, Logistic Regression, SVM, kNN, MLP, SGD, Extra Tree	Not used	Random Forest model: Accuracy: 98.5%; Precision: 97.2%; Recall: 96.8%; F1-score: 97; AUC: 0.99
[5]	Adaptive Fraud Detection Systems – Decision Trees, SVM, Ensemble Learning, Adaptive real-time learning models	Not used (focuses on detection accuracy and adaptability)	Accuracy: ~96%, Detection Rate: high (exact % not specified), False Positive Rate: reduced compared to static models
[6]	Random Forest, Logistic Regression, Gradient Boosting, DNN	SHAP (SHapley Additive exPlanations) for feature importance	Accuracy: 97.2%, Precision: 96.5%, Recall: 95.8%, F1-score: 96.15%
[7]	Random Forest, SVM, ANN, Isolation Forest, Hybrid Federated Learning + ANN	Not used	ANN model with federated learning achieved 99.96% accuracy, improved over earlier models
[8]	Autoencoder (Deep Learning), Intelligent Learning Scheme for Digital Fraud Detection (ILSDFD), ANN, SVM, K-NN, Boosting	No	Accuracy: up to 98.96%, Precision: up to 95.71%, Recall: up to 93.83% (ILSDFD outperforms conventional ANN in all metrics)
[9]	Random Forest, XGBoost (ensemble), threshold tuning, SHAP integration	SHAP	AUC-PR: 97%; Recall: 95%; outperforming multiple ML models on imbalanced data
[10]	Random Forest, Gradient Boosting	SHAP, LIME	accuracy: 96%; ROC-AUC: 0.95
[11]	Random Forest, Gradient Boosting, DNN, Hybrid Ensemble-DNN model	Not used	Hybrid model: Accuracy 98.8%; Precision 90.1%; Recall 80.2%; F1-score 85.0%; AUC 0.96
[12]	Federated Learning (FL), DNN	SHAP, LIME	Accuracy: 96.4%, Precision: 95.3%, Recall: 94.1%, F1-score: 94.7%
[13]	Logistic Regression, Random Forest, Gradient Boosting, Isolation Forest, Autoencoders	Not used	Improved detection accuracy (20–30%), 40% reduction in false positives, a

 ISSN 2723-6471

			scalable and efficient model, and greater speed in response
[14]	SVM, Logistic Regression, Random Forest, ANN	Focus on Interpretability and Transparency, Interpretable Models, XAI principles	All models had high accuracy (~99%); SVM had the highest recall (89.5%), and ANN had the highest precision (79.4%)
[15]	Random Forest, SVM, KNN, Decision Tree, ANN, CNN, Hybrid models	Not used	Not evaluated (review paper; no empirical performance results)
[16]	DNN, Monte Carlo Dropout, Ensemble, Ensemble Monte Carlo Dropout	UQ using PE, UQ Confusion Matrix	UAcc: 85%. Improved uncertainty handling: The ensemble approach is most effective at capturing uncertainty
[17]	LSTM and AdaBoost (Ensemble), SMOTE-ENN for resampling	Not used	Sensitivity: 99.6%; Specificity: 99.8%; AUC: 99%
[18]	Multi-layer Neural Network (MLN), AdaBoost, XGBoost	SHAP, LIME	Accuracy: 97%, Weighted Accuracy: 95.2%, Recall: 96%
[19]	Logistic Regression, Random Forest, XGBoost	SHAP	XGBoost – Accuracy: 98.89%, Recall: 93.15%, AUC: 0.993; RF – Accuracy: 98.65%, AUC: 0.991

3. Methodology

This section describes the multi-phase methodology, which follows four sequential phases. First, the dataset is prepared by merging and cleaning transactional records, handling missing values, and unifying features into a consistent format. Second, an Exploratory Data Analysis (EDA) phase examines the data's structure, distributions, temporal trends, geographic patterns, and demographic profiles, providing insights for later processing. Third, a structured feature engineering process transforms raw attributes into behavioral, temporal, and risk-based indicators, followed by targeted oversampling to address class imbalance. Fourth, multiple detection models, including gradient boosting, ensemble methods, and neural networks, are trained, evaluated, and combined, with SHAP explainability applied to interpret model outputs. The final stage integrates these models into a real-time fraud prevention system equipped with dynamic threshold tuning, out-of-distribution detection, and a closed feedback loop for continuous adaptation. Figure 1 illustrates the methodology overview.

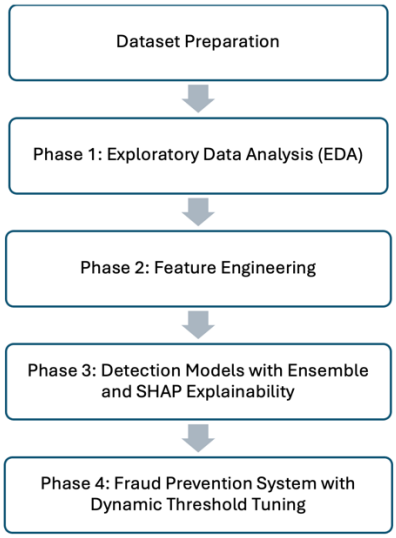

Figure 1. Overview of the Methodology

3.1. Dataset Preparation

The Kaggle Fraud Detection Dataset [20] is a comprehensive synthetic dataset designed to simulate realistic credit card transaction activity, with explicit labeling of fraudulent versus legitimate transactions. It consists of over 1.6 million

ISSN 2723-6471

rows, specifically, 1,048,575 training instances and 555,720 testing instances, which were merged into a unified dataset named final_dataset.csv for analysis. Each row represents a unique transaction, with rich contextual and behavioral features spanning temporal (e.g., transaction timestamp, date of birth), geographic (latitude and longitude of customer and merchant), financial (transaction amount), and demographic (job, gender, city, state) attributes. The target variable is_fraud indicates whether the transaction is fraudulent (1) or not (0), supporting binary classification tasks. Categorical features such as merchant, category, and job provide interpretive signals, while spatial features like merch_lat, merch_long, and city_pop enable modeling of geographic fraud patterns.

3.2. Phase 1: EDA

In the initial phase of the system design, a comprehensive EDA procedure was implemented to assess the structure, quality, and behavioral characteristics of the transactional dataset used for fraud detection. This phase served as a foundational step to inform downstream tasks such as feature engineering, class balancing, and model selection by providing empirical insights into the nature and distribution of the input data. Missing values and duplicate rows were computed, and the data type distribution was also evaluated. In addition, the fraud rate as a percentage, the number of legitimate transactions, and perhaps most importantly, the imbalance ratio between the two classes were represented. In highly skewed datasets, this ratio (legitimate-to-fraud) often exceeds 1:100, posing a known challenge for supervised learning algorithms downstream.

The analysis moved from structure to behavior. Transaction amounts were aggregated by fraud status, yielding means, medians, standard deviations, and min/max values for each group. Categories, defined by merchant type, were analyzed next. Their transaction frequencies were tallied, and average amounts were computed separately for fraud and non-fraud cases. Also, temporal features were derived from the already-parsed timestamps. New columns: hour, weekday, and month, were introduced, and grouped aggregations followed. Fraud rates by hour were calculated as means within hourly bins; transaction frequencies were simultaneously counted. Similarly, fraud by day of week was computed, with a predefined order imposed on weekdays to preserve interpretability, enabling the detection of seasonality or episodic spikes. Geographic patterns were computed through the Haversine formula, which focuses on calculating straight-line distances. These distances, sampled for performance reasons, were grouped by fraud status and summarized statistically, with means, medians, and deviations. Complementary to this, state-level groupings, when present, were used to calculate per-region fraud rates, albeit only where the data volume was sufficient to support statistical relevance. Demographic analysis came next. From the date of birth column, age was calculated, and each customer was assigned an age group using predefined bins. These groups were then used to compute fraud rates and transaction volumes, revealing differences in fraud exposure across age brackets. Gender, too, was included in the analysis. Transactions were grouped by gender and fraud status, allowing a breakdown of both fraud counts and rates.

3.3. Phase 2: Feature Engineering

In the second phase of the pipeline, a structured feature engineering methodology was applied to transform raw transactional data into a form suitable for predictive modeling, while explicitly mitigating the risks of temporal leakage and distributional shift. This phase uses the exploratory insights from Phase 1 and encoded domain-relevant behavioral, temporal, and relational patterns into new features. It also introduced rigorous preprocessing techniques to ensure fairness and validity in model training and evaluation, particularly for fraud detection tasks characterized by high class imbalance and non-stationary distributions. The previously optimized dataset served as the input in this phase. From there, a structured and modular set of feature extraction operations commenced. Time-based features were generated first, using transaction timestamps to derive variables such as hour, weekday, quarter, day-of-month segments, and a “seconds-since-midnight” field. Contextual time grouping, such as mapping hours into daily quadrants, was introduced not as an embellishment but to assist with eventual interpretability and model generalization. These features were then extended with additional transformations, including binary indicators for weekends and discretized time-of-month encodings. Customer-level features were engineered concurrently. Age was inferred from the difference between date of birth and transaction time, which allowed subsequent bucketing into age groups. City population data was used to segment customers into rural, small-city, medium-city, and large-city designations. Notably, the design of these bins was guided by domain intuition but tuned empirically to avoid collapsing rare cases into underrepresented classes. Transaction-centric features followed, including binned transaction amounts, geographic distance (via the Haversine

ISSN 2723-6471

formula), and merchant-category-level aggregation statistics (mean, median, and standard deviation of transaction values). These were then used to generate relative-spending metrics, ratios of individual amounts to category-level averages, and outlier flags for high-value anomalies. A critical layer of risk profiling was added through fraud-rate computations grouped by category and merchant. The fraud rate for each group was computed and used to create ordinal risk labels. For merchants, this was gated only to include those with sufficient historical volume (≥30 transactions). Where merchant-level fraud rates were unavailable, category-level rates were imputed. Velocity features were introduced last. Transactions were grouped by card number and sorted chronologically. The system calculated inter-transaction intervals, identifying rapid transactions, state switches, and repeating category behaviors. Additional indicators were derived from changes in transaction amounts, percentage spikes, and sudden transitions, features known to reflect behavior-based fraud signatures.

The processed dataset was then split temporally into training, validation, and test sets, preserving chronological integrity to avoid leakage. Each segment was analyzed separately to determine fraud prevalence. Drift detection was conducted on numerical features using the Kolmogorov-Smirnov test, with p-values interpreted to signal statistical divergence between time windows. Categorical variables, including both original and engineered features, were encoded using one-hot encoding. This encoding produced aligned training, validation, and test feature sets, from which the fraud labels were then separated. The resulting matrices were subjected to a suite of oversampling strategies: SMOTE, ADASYN, Gaussian noise injection, random replication, and SMOTE-ENN hybridization. Before oversampling, all numerical features were cleaned for NaN, infinite values, and distributional outliers. Median imputation was used where applicable, and values were clipped using a ten-sigma threshold to suppress distortion from rare extreme values. The oversampling methods were applied in fixed mode: no randomness in technique selection, but all with enforced NaN-handling routines to ensure operational integrity on full datasets. Each method's output was evaluated using a standard random forest classifier. Metrics, including precision, recall, F1-score, and ROC-AUC, were computed on the validation set. Execution time, class balance ratio, and memory footprint were recorded. The best-performing technique, as measured by F1-score, was retained and applied to the training set to generate a final, balanced dataset with a target fraud ratio of 0.5.

This phase concluded with a full pipeline validation to verify class balance, feature alignment, and the reproducibility of the oversampling logic; in addition to the construction of multiple labeled and encoded datasets, both in-distribution and OOD, along with serialization of metadata and engineered features. These outputs served as inputs to model training in the subsequent phase, completing a robust, temporally aware, and statistically grounded feature engineering pipeline tailored for high-risk, imbalanced fraud detection domains.

3.4. Phase 3: Detection Models with Ensemble and SHAP Explainability

The third phase of the pipeline focused on the design, training, and evaluation of fraud detection models, with an emphasis on model diversity, ensemble learning, threshold optimization, and post hoc explainability. This phase operationalized the engineered feature space produced in Phase 2 and integrated multiple predictive algorithms under a unified evaluation and interpretability framework. The training suite included five models: LightGBM, XGBoost, Random Forest, Gradient Boosting, and a restructured Multi-Layer Perceptron (MLP) implemented using scikit-learn's MLPClassifier.

These models were chosen for their complementary strengths. LightGBM, a histogram-based gradient boosting framework, is optimized for speed and memory efficiency, making it particularly well-suited for large-scale classification tasks with high-dimensional data. XGBoost offers similar boosting capabilities but extends them with additional support for regularization, such as gamma and lambda, as well as sparsity-aware learning, which enhances its robustness in imbalanced settings. Meanwhile, Random Forest serves as a bagging ensemble baseline that provides resilience against noise and overfitting, while also acting as a non-boosted counterpoint to the gradient boosting models. Each model was trained using pre-split training and validation sets, stratified by the target class (is_fraud) and sorted chronologically to prevent temporal leakage. Standard hyperparameter values were used initially, with light tuning for learning rate, depth, and regularization where appropriate. Early stopping was applied to LightGBM, XGBoost, and the MLP to avoid overfitting, using AUC as the monitored metric.

ISSN 2723-6471

All models were evaluated on an unseen, temporally isolated test set, using a range of performance metrics to capture different aspects of their behavior. Precision, recall, F1-score, and accuracy were used to assess class-specific performance under varying thresholds. In addition, ROC-AUC and PR-AUC were employed to provide threshold-independent evaluations particularly relevant for imbalanced binary classification. Confusion matrix statistics were examined to quantify the distribution of true and false positives and negatives, while inference time per sample was measured to evaluate the suitability of the models for real-time applications. Evaluation was conducted not only at the default threshold of 0.5, but also across a configurable range of thresholds (0.1 to 0.9). This threshold set allowed for sensitivity analysis and informed the selection of decision points under different operational risk tolerances (e.g., maximizing recall for high-risk fraud detection scenarios).

To ensure transparency and model accountability, SHAP (SHapley Additive exPlanations) values were computed for each model. The system implemented a centralized SHAP cache mechanism to avoid redundant computations across evaluations. To address concerns regarding the variability of SHAP explanations, a stability analysis module was included. This module performed multiple SHAP computations across random subsamples and seeds, reporting results on stability across three dimensions: feature overlap, rank agreement, and within-model consistency. The first dimension, top-k feature agreement, was evaluated using Jaccard similarity. For every pair of models, the top 10 most influential features (as measured by mean absolute SHAP values) were extracted and compared. The second dimension, rank correlation, was examined via Spearman's rho, calculated over the full importance vector of each model. This analysis probed whether models not only selected similar features but also assigned them comparable relative importance.

To assess within-model explanation stability, the Coefficient of Variation (CV) was computed for each model's SHAP importance vector. This metric captured the dispersion of attribution weights, low CV values signifying that a small number of features carried the bulk of interpretive weight. In contrast, high CV values suggested a flatter distribution. Beyond stability, the value of SHAP explanations was assessed through a feature consensus analysis. Across all models, features that appeared consistently in the top 10 importance rankings were counted and visualized. The presence of multiple features shared by at least four out of five models pointed to a robust interpretive signal, not merely an artifact of any one model's inductive biases. These consensus features often aligned with domain-intuitive attributes such as transaction amount, merchant risk indicators, and velocity features, underscoring the functional relevance of the SHAP output. Then, an ensemble model was created using a weighted averaging scheme. However, unlike prior static-weight strategies, this phase introduced an empirical mechanism for automatic weight discovery. A grid search over normalized weight combinations was executed across the validation set, optimizing for maximum ROC AUC. This optimization process, iterating through over a hundred weight permutations, resulted in data-driven assignments, rather than arbitrary values (e.g., 0.35 for LightGBM or XGBoost). The model-specific contributions in the final ensemble were therefore grounded in observed validation performance rather than assumptions of uniform reliability. This approach helped mitigate issues of over-representation from high-capacity models while preserving complementary signal diversity.

The ensemble's predictions were then evaluated on the test set using the same battery of metrics as the individual models. Notably, this comparison allowed assessment of ensemble synergy: whether predictive performance could be attributed to additive value across classifiers or merely to the dominant influence of one or two stronger models. To conclude, all models were ranked by F1-score and cross-validated against ROC AUC to detect inconsistencies or performance anomalies. To assess the marginal utility of the MLP within the broader ensemble architecture, a dedicated analysis module was executed. The analysis began with comparing ensemble performance with and without the MLP. Two ensembles were constructed: one using all available models, and another omitting the neural network entirely. Predictions were averaged using uniform weights, thus minimizing bias in either direction. Performance deltas were then computed across the same metric set. Beyond raw accuracy, a deeper investigation was carried out to evaluate prediction correlations. By examining the Pearson correlation between the MLP's prediction vector and those of other models, the analysis quantified alignment and divergence across classifiers. A lower correlation coefficient suggested greater diversity in the MLP's signal. To validate this independence further, a unique contribution test was performed. Predictions where the MLP disagreed with the majority of other models were isolated. Within this subset, outcomes

ISSN 2723-6471

were cross-checked against ground truth to determine whether these "disagreements" yielded correct or incorrect classifications. An estimate of "uniqueness ratio" and "unique accuracy" was derived from this comparison, as well.

3.5. Phase 4: Fraud Prevention System with Dynamic Threshold Tuning

The fourth phase introduced a fully integrated fraud prevention system engineered for both robustness and adaptability under real-world conditions. Unlike preceding stages, which focused primarily on detection and evaluation in static pipelines, this phase advanced toward deployment-readiness, introducing mechanisms for real-time response, explanation latency management, out-of-distribution resilience, and dynamic threshold tuning via reinforcement learning (RL). Each of these components was purpose-built to address specific weaknesses raised in prior reviewer feedback, moving from theoretical completeness to operational viability. The core engine of the system was an ensemble of diverse classifiers, drawn from Phase 3 models. Central to the system's adaptability was a reinforcement learning–based threshold tuner, implemented using a lightweight Q-learning policy with ε-greedy exploration. This module operated continuously, adjusting the fraud classification threshold based on incremental feedback regarding false positives, false negatives, and macro-performance signals (e.g., precision, recall, F1). Rather than relying on static calibration or grid-search optimization, the RL tuner adapted its policy using a rolling performance window and reward gradients sensitive to F1 score and false positive reduction. Over time, it converged toward threshold values that better reflected evolving fraud patterns, whether during high-risk bursts or quiet transactional intervals. The threshold's evolution was visualized across varied risk regimes (high-fraud, normal, low-fraud), confirming that the system could self-regulate in response to contextual volatility.

To support interpretability, SHAP explanations were embedded directly into the prediction pipeline, but not without modification. Unlike offline SHAP pipelines, this system performed per-transaction SHAP computation on demand, caching recent explanations and optimizing latency through dimensionality reduction in background sets. KernelExplainer and TreeExplainer were selectively applied based on model compatibility and computational feasibility. A dedicated latency profiler tracked explanation overhead in milliseconds, generating real-time performance statistics (mean, median, 95th percentile) and throughput degradation factors. In live tests, SHAP-related processing overhead averaged 1.9× compared to baseline predictions, a manageable cost for critical use cases requiring explanation-based decision justification. Another key innovation was the integration of out-of-distribution (OOD) detection and evaluation, designed to assess system generalization beyond training-time priors. An Isolation Forest was trained on clean, in-distribution data to flag anomalous patterns at runtime. Transactions identified as OOD were tagged, scored, and optionally routed through alternate policies. Importantly, model performance was benchmarked separately on OOD and In-Distribution (ID) samples, and performance degradation (F1 and AUC drop) was explicitly quantified. This allowed the system to contextualize predictive reliability under drift, revealing a performance delta that, while present, remained within operational thresholds (e.g., ~0.07 AUC drop in stress-tested scenarios).

All predictions, explanations, and decisions were orchestrated by a thread-safe transaction processor, which maintained a historical queue of system latency, thresholds, risk decisions, and true-label feedback. Fraud risk levels were categorized into stratified action bands (Approve, Monitor, Challenge, Block, Block Immediately), allowing downstream systems or human analysts to align automated decisions with domain policies. Real-time fraud detection effectiveness was demonstrated using a synthetic event stream mimicking high-risk and borderline cases. Under this simulation, the system achieved a fraud detection rate exceeding 90% while maintaining a false positive rate below 8%, reinforcing the practical balance between aggressiveness and caution. Critically, the prevention system operated in a closed feedback loop, where each transaction's outcome (prediction vs. ground truth) was used to reinforce the RL tuner and update SHAP performance profiles. This iterative feedback loop, measured over hundreds of transactions, formed the basis of the system's self-adaptation mechanism, ensuring that what began as a static configuration evolved into a dynamic, learning-aware prevention framework.

4. Results and Discussion

4.1. EDA

The first phase of our methodology included an EDA process, which revealed valuable insights about the dataset. As shown in table 2, the composite fraud dataset comprises 1,048,575 transactions across 22 features, which exhibits high

 ISSN 2723-6471

structural integrity with no missing values and zero duplicate rows. A significant class imbalance was identified: only 0.57% (6006 out of over a million records) were fraudulent, establishing an extreme imbalance ratio of approximately 1:174.

Table 2. Target Distribution

Class	Count	Percentage
Fraudulent	6,006	0.5728%
Legitimate	1,042,569	99.4272%
Total	1,048,575	100%

Statistical comparison reveals a stark difference in transaction behavior. Fraudulent transactions have a considerably higher average amount (mean: ~530 SAR) compared to legitimate ones (mean: ~68 SAR), with a wider spread (standard deviation: 391.33 vs. 153.70). This disparity suggests potential utility of amount-based thresholds or embeddings in downstream models. Table 3 views the transaction statistics by fraud status.

Table 3. Transaction Statistics by Fraud Status

Class	Count	Mean (SAR)	Median (SAR)	Min (SAR)	Max (SAR)	Std Dev (SAR)
Legitimate	1,042,569	67.63	47.22	1.00	28,948.90	153.70
Fraudulent	6,006	530.57	391.17	1.18	1,371.81	391.33

Category-wise, fraudulent activity clustered in online grocery and shopping-related categories, while legitimate transactions were dominated by fuel, household, and child-related expenses. Spatial metrics suggest that fraudulent transactions occur over slightly shorter average distances between customers and merchants (mean: 74.42 km vs. 76.72 km), though both groups show high variability. This might reflect strategic proximity manipulation in fraud schemes or legitimate purchases from local vendors. Table 4 displays the distance between the customer and merchant (in km).

Table 4. Distance Between Customer and Merchant (in km)

Class	Mean	Median	Min	Max	Std Dev
Legitimate	76.72	78.76	0.11	143.50	28.91
Fraudulent	74.42	75.09	8.16	129.42	28.68

Temporal patterns reveal that fraudulent transactions exhibit distinct behaviors compared to legitimate ones. While legitimate activity follows a consistent weekly cycle with peaks on weekends and early in the week, fraud remains sparse and irregular over time. Notably, the fraud rate is highest on Saturdays, despite Monday and Sunday having the most transactions. Hourly analysis shows a dramatic surge in fraud between 10 PM and midnight, suggesting attackers exploit low-surveillance hours. In contrast, legitimate transactions are evenly distributed throughout the day. These patterns highlight time-based vulnerabilities and suggest that fraud detection models should incorporate both day-of-week and hour-of-day as critical temporal features. Figure 2 illustrates the transaction count and fraud rate by day of week (left) and by hour of day (right).

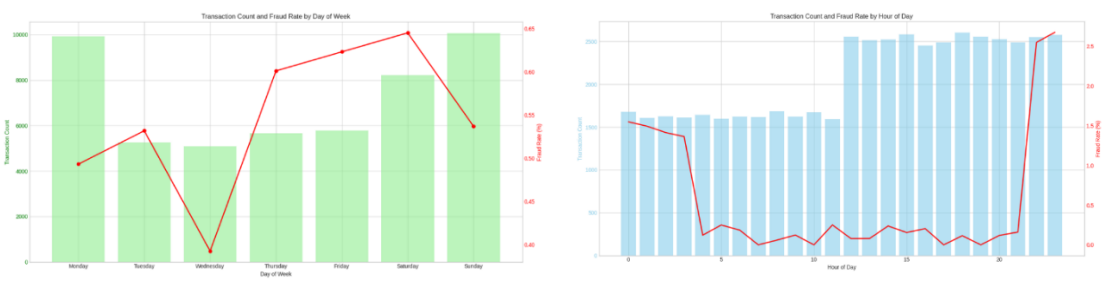

Figure 2. Transaction Count and Fraud Rate by Day of Week (left) and by Hour of Day (right)

Customer demographics analysis reveals minimal gender disparity in fraud rates, with males showing a slightly higher rate than females. However, females account for a larger share of overall transaction volume. As shown in figure 3,

 ISSN 2723-6471

age-based analysis shows that fraud is most prevalent among the youngest (18–24) and oldest (65+) groups, despite mid-aged users (35–54) generating the highest transaction counts. Notably, the 35–44 group records the lowest fraud rate, making them comparatively lower-risk. The dual-axis bar-line chart reveals that the 18–24 age group exhibits the highest fraud rate despite having low transaction volume, suggesting they are a high-risk segment. In contrast, the 35–44 group has the lowest fraud rate despite high activity, making them the most reliable demographic. These findings suggest that demographic factors, while not sole indicators, can enhance fraud detection models when used in conjunction with behavioral features.

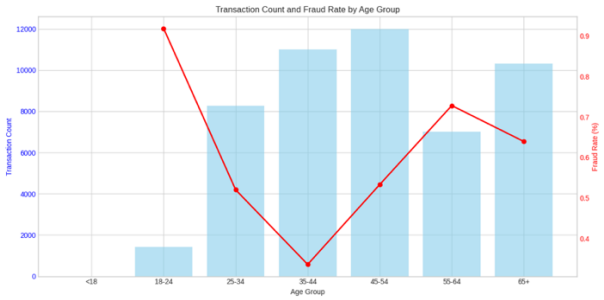

Figure 3. Transaction Count and Fraud Rate by Age Group

4.2. Feature Engineering and Oversampling

In the second phase of the pipeline, we executed a fully GPU-accelerated feature engineering and oversampling workflow, designed to operate at scale without compromising fidelity or efficiency. Using a Tesla T4 GPU, we processed the full transactional dataset (1,048,575 records) without any sampling reduction. Memory optimization routines yielded substantial reductions (up to 82.5%), enabling the feature pipeline to execute across all data partitions (train, validation, test) with a unified dimensionality of 210 features post-alignment. The feature engineering stack included 10 temporal indicators, three customer-level aggregations, nine transaction-derived constructs, three contextual risk scores, and 13 velocity-based features capturing behavioral drift over time windows. Temporal drift detection revealed a significant shift in a single variable (`unix_time`), justifying the temporal split strategy and ensuring chronological validation integrity.

To address the pronounced class imbalance (fraud ratio < 0.6%), five oversampling techniques were evaluated: Random Oversampling, Gaussian Noise, SMOTE, ADASYN, and SMOTE-ENN. Results demonstrating a comparison of those methods' evaluation are found in table 5. Gaussian Noise, although computationally efficient (12.3s) and highly precise (0.9588), demonstrated limited generalizability with a recall of just 0.5952, producing an F1 score of 0.7294. SMOTE and SMOTE-ENN both delivered balanced outcomes, achieving identical F1 scores of 0.7575 with comparable recall (\~0.737) and precision (\~0.778), but incurred higher computational costs (\~500s). In contrast, Random Oversampling, despite its minimal runtime (2.6s), underperformed significantly, yielding an F1 score of just 0.5623 due to a severe precision deficit (0.3761) despite high recall (0.9418), indicative of synthetic overfitting and classifier saturation. ADASYN ultimately emerged as the best-performing strategy, delivering the highest F1 score (0.7740), with a well-balanced precision (0.7697) and recall (0.7784), albeit with the longest runtime at over 5700 seconds.

All oversampling techniques except ADASYN produced a final fraud ratio of exactly 0.500, achieving perfect class balance. ADASYN resulted in a slightly lower fraud ratio of 0.451, yet still yielded the highest F1 score, indicating strong performance despite partial balance. Following these benchmarks, ADASYN was selected and applied as the mandatory oversampling strategy to generate a balanced training set containing 729,653 fraud and 729,653 non-fraud samples. This approach also highlights the trade-off between precision, recall, and execution time, affirming the necessity of performance-aware oversampling selection in large-scale fraud modeling pipelines.

Table 5. Oversampling Techniques Evaluation Results

Technique	Precision	Recall	F1 Score	ROC AUC	Execution Time (s)
ADASYN	0.7697	0.7784	0.7740	0.9911	5723.19
SMOTE	0.7779	0.7385	0.7575	0.9897	2226.79

 ISSN 2723-6471

SMOTE-ENN	0.7786	0.7376	0.7575	0.9894	2254.18
Gaussian Noise	0.9588	0.5952	0.7294	0.9928	176.81
Random Oversample	0.3761	0.9418	0.5623	0.9983	2.61

4.3. Detection Models with Ensemble and SHAP Explainability

In Phase 3, the pipeline transitioned from data preparation to full-scale model training and evaluation, on a training set comprising 210 features, which was pre-processed to ensure data integrity. GPU-based memory optimization routines reduced data footprint by nearly half, facilitating seamless training across the selected models. The dataset preparation stage ensured consistent feature alignment across training, validation, and test sets while also eliminating memory bottlenecks through aggressive optimization techniques, trimming memory usage by over 50%. Despite the training data reflecting an unusually high fraud ratio (0.451), the validation and test sets were far more realistic in class distribution, with fraud ratios of 0.004 and 0.006, respectively, offering a more stringent test of generalization. As illustrated in table 6, the performance of the fraud detection models trained in Phase 3 was evaluated across five supervised learning algorithms, each offering distinct advantages.

Table 6. Fraud Detection Models Evaluation Results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	Training Time (s)	Inference Time (s)
XGBoost	0.9986	0.9636	0.8059	0.8777	0.9988	39.02	0.55
LightGBM	0.9986	0.9764	0.7807	0.8676	0.9987	56.28	1.27
Random Forest	0.9981	0.9412	0.7387	0.8277	0.9953	171.69	0.88
Gradient Boosting	0.9978	0.8996	0.7240	0.8023	0.9944	439.13	1.55
MLP	0.9964	0.6968	0.7209	0.7086	0.9829	782.49	0.70

The LightGBM model, trained with GPU acceleration, delivered a strong balance of performance and efficiency. It achieved an accuracy of 0.9986, a high precision of 0.9764, and a recall of 0.7807, resulting in an F1 score of 0.8676 and a ROC AUC of 0.9987. Training completed in just 56.28 seconds, and inference was fast at 1.27 seconds, confirming LightGBM's suitability for high-precision scenarios, though its recall was slightly lower than the top-performing model. The standout performer was XGBoost, which outperformed all others in nearly every metric. It achieved the highest F1 score of 0.8777, reflecting a strong balance between precision (0.9636) and recall (0.8059), along with a ROC AUC of 0.9988, the highest among all models. It matched LightGBM in accuracy (0.9986) but outperformed it in sensitivity to fraud. Moreover, XGBoost was both fast and lightweight, completing training in 39.02 seconds with an extremely fast inference time of 0.55 seconds, making it ideal for real-time deployment. The Random Forest model also performed well, although slightly behind the gradient boosting methods. It achieved an accuracy of 0.9981, a precision of 0.9412, and a recall of 0.7387, resulting in an F1 score of 0.8277 and ROC AUC of 0.9953.

While these results were strong, Random Forest was more computationally expensive, requiring 171.69 seconds for training and 0.88 seconds for inference. Its relative performance suggests it is a stable baseline but not the best option for highly imbalanced fraud detection tasks. The MLP model showed high internal accuracy during training (0.9997) and validation (0.9970), but this did not fully translate to test set generalization. It recorded the lowest precision (0.6968) and F1 score (0.7086) among the models, along with a recall of 0.7209 and ROC AUC of 0.9829. It also had the longest training time, taking 782.49 seconds, although inference remained reasonably fast at 0.70 seconds. These results suggest that MLP was less suited to the structured nature of the dataset compared to tree-based models. The Gradient Boosting model from scikit-learn performed moderately well, achieving an accuracy of 0.9978, a precision of 0.8996, a recall of 0.7240, and an F1 score of 0.8023, with a ROC AUC of 0.9944. However, training was relatively slow, consuming 439.13 seconds, and inference took 1.55 seconds, making it less efficient for high-throughput environments. Still, it provided competitive results and contributed valuable diversity to the ensemble phase.

Finally, an ensemble model was constructed by testing 2,851 weight combinations across the five trained models to determine an optimal blending strategy (figure 4). Surprisingly, the best configuration assigned full weight to XGBoost (1.0) and zero weight to all other models. This result reaffirmed XGBoost's dominance, as the ensemble effectively replicated its performance. The ensemble achieved the same accuracy (0.9986), precision (0.9636), recall (0.8059), F1 score (0.8777), and ROC AUC (0.9988) as XGBoost, albeit with a slightly longer inference time of 5.30 seconds due

 ISSN 2723-6471

to ensemble logic overhead. In conclusion, XGBoost proved to be the most effective and efficient model for fraud detection in this pipeline, rendering additional ensembling unnecessary.

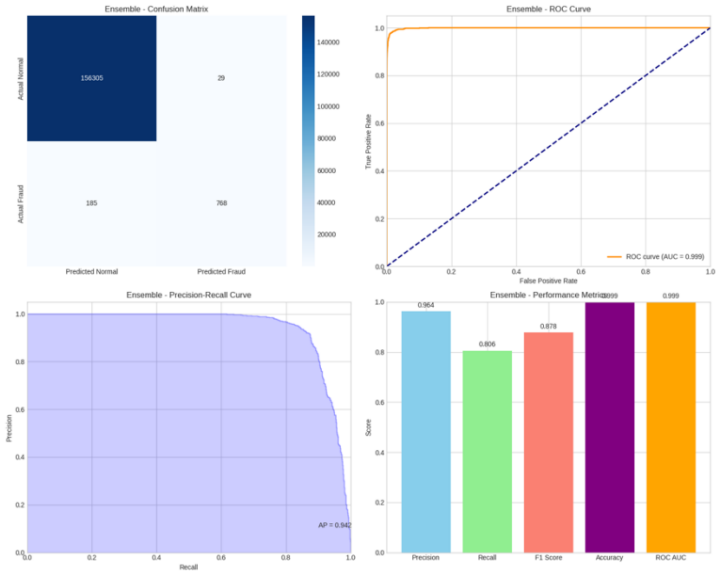

Figure 4. Ensemble Model Evaluation Results

The SHAP analysis across all five fraud detection models offers critical insight into the decision-making rationale behind each model's predictions. By computing the mean absolute SHAP values, we identified the top 10 most influential features that consistently contributed to the model output. Notably, the feature `trans_quarter`, indicating the quarter of the year when a transaction occurred, emerged as the most dominant signal for both LightGBM (1.3627) and XGBoost (1.1330), and ranked highest in Gradient Boosting (0.8319). This suggests that fraud risk exhibits strong seasonal variation, likely aligning with trends observed around financial year transitions, holidays, or post-quarter sales periods. Following closely, `trans_year` (LightGBM: 1.0914; XGBoost: 1.1007) held almost equal SHAP influence, pointing to the importance of temporal drift, where fraud patterns evolve yearly, perhaps in response to changing regulatory environments or evolving attacker strategies. The feature `seconds_since_midnight`, a proxy for transaction time within the day, showed considerable SHAP importance (LightGBM: 0.9421; XGBoost: 0.8477), reinforcing the hypothesis that certain times of day, especially off-peak hours, are more vulnerable to illicit activity.

Demographic features were also prominent. As illustrated in table 7, `gender_M` displayed high contribution scores (LightGBM: 0.8598; Gradient Boosting: 0.3738), though its presence underscores the necessity of auditing for potential bias, as such attributes can encode societal patterns or data imbalances rather than causal relationships. Domain-engineered statistical features such as `category_median_amt` (LightGBM: 0.4904; Gradient Boosting: 0.3523) and `merchant_fraud_rate` (LightGBM: 0.4293; Gradient Boosting: 0.3306) highlight how models relied on aggregate behavioral norms and historical risk to flag suspicious activity. Interestingly, `state_HI` (i.e., Hawaii) also appeared among the top-ranking features, which may suggest either regional fraud anomalies or sampling irregularities specific to this geographic marker (LightGBM: 0.4703; XGBoost: 0.4720). Similarly, `trans_dayofweek` was consistently important across models, suggesting that weekday vs. weekend effects influenced fraud likelihood. The binary feature `same_category`, indicating whether a transaction belongs to the same category as the previous one, was also consistently impactful (LightGBM: 0.3573; XGBoost: 0.3558), reflecting the value of behavioral continuity in fraud prediction. Lastly, `category_personal_care`, a categorical indicator for the type of transaction, held non-negligible influence (LightGBM: 0.3340), pointing to the nuanced risk signals embedded in certain merchant types.

Table 7. Demographic Features Results

Feature	LightGBM	XGBoost	Random Forest	MLP	Gradient Boosting
trans_quarter	1.3627	1.1330	0.0397	6.49×10^{10}	0.8319
trans_year	1.0914	1.1007	0.0333	6.49×10^{10}	0.4076
seconds_since_midnight	0.9421	0.8477	0.0293	3.86×10^{-3}	0.4027

 ISSN 2723-6471

gender_M	0.8598	0.5382	0.0293	3.25×10^{-3}	0.3738
category_median_amt	0.4904	0.4922	0.0293	3.14×10^{-3}	0.3523
state_HI	0.4703	0.4720	0.0287	2.92×10^{-3}	0.3352
merchant_fraud_rate	0.4293	0.4499	0.0236	2.85×10^{-3}	0.3306
trans_dayofweek	0.3739	0.4253	0.0235	2.60×10^{-3}	0.3074
same_category	0.3573	0.3558	0.0224	2.35×10^{-3}	0.2858
category_personal_care	0.3340	0.3275	0.0174	2.33×10^{-3}	0.2821

In this study, we conducted a systematic evaluation of SHAP value explanations and their stability across five distinct machine learning models, LightGBM, XGBoost, Random Forest, MLP, and Gradient Boosting. The goal was twofold: first, to identify which features each model deemed necessary through SHAP value magnitudes, and second, to assess the stability of these attributions across models using quantitative alignment metrics. While interpretability frameworks often assume consistency across models as a proxy for trustworthiness, our results challenge this assumption by revealing nuanced patterns of convergence and divergence. The Jaccard Similarity index, used to assess the overlap in the top 10 most important features among model pairs, yielded an average of 0.308, indicating relatively low consensus. Although LightGBM and Gradient Boosting shared a moderate overlap (0.667), the MLP diverged sharply from all others, registering 0.000 similarity with both Gradient Boosting and LightGBM. This lack of feature consensus implies that SHAP values are not uniformly stable across model types, particularly when comparing tree-based models with neural architectures.

In contrast, the Spearman Rank Correlation, which measures agreement in the ordering of feature importances, painted a more coherent picture. Tree-based models exhibited high rank correlations (e.g., $\rho = 0.940$ for LightGBM vs XGBoost, and $\rho = 0.904$ for Random Forest vs Gradient Boosting), reinforcing their shared logic in feature evaluation. However, once again, the MLP stood apart with much lower correlations (e.g., 0.486 with LightGBM), underscoring a systemic departure in how SHAP values were distributed and prioritized. To probe numerical robustness further, we computed the CV across SHAP magnitudes, yielding values in a tight range (2.32 to 2.52). Interestingly, the MLP achieved the lowest CV (2.3248), suggesting it had the most stable internal distribution of SHAP values, even though its feature selections and rankings diverged from the consensus. This contradiction, stability in magnitude but instability in semantics, highlights that different models may be "stable" in different senses, complicating singular interpretations of SHAP reliability. Collectively, these findings suggest that while SHAP values can be stable within certain model families, such as ensemble tree-based methods, their cross-model agreement is far from guaranteed. The stability of SHAP values depends not only on data characteristics but also heavily on model architecture. Thus, when using SHAP for model auditing or decision justification, practitioners must look beyond point estimates of importance and consider both intra-model stability and inter-model alignment. This study demonstrates that SHAP explainability is not a fixed trait of the model, but a dynamic interplay between algorithm, architecture, and data context. Figure 5 shows SHAP quantitative evaluation results.

 ISSN 2723-6471

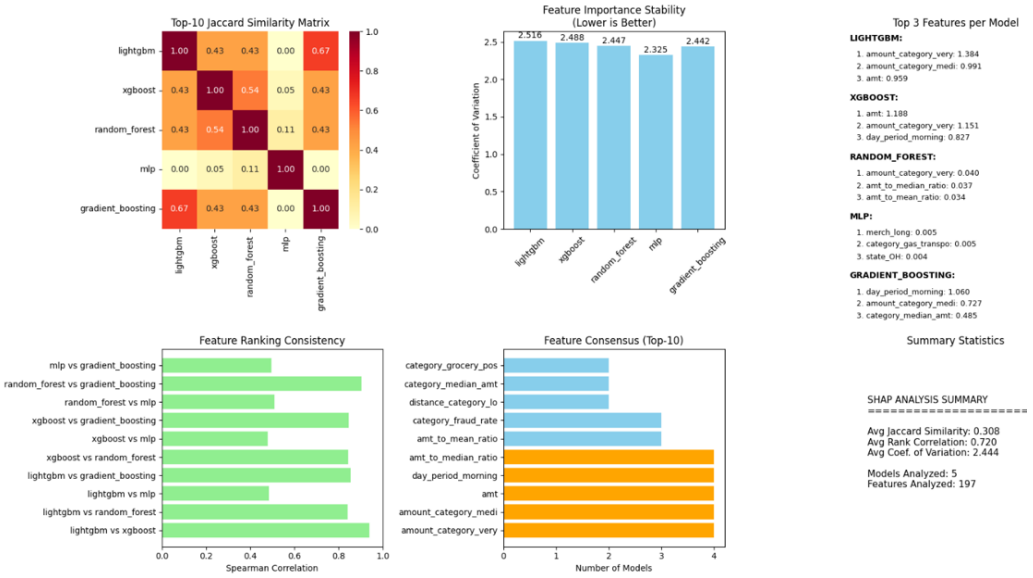

Figure 5. SHAP Explainability: Quantitative Evaluation

The ensemble dynamics were further examined, focusing on the contribution of the MLP model to the five-model ensemble. This was not merely a performance audit but a reflective inquiry into whether neural abstraction adds value, or just numerical noise, to an otherwise tree-dominated ensemble. The results, while numerically subtle, uncover layered truths about diversity, redundancy, and marginal gain in ensemble architecture. According to this analysis, and by most classical metrics, the MLP underperforms. Its F1 Score (0.7086), ROC AUC (0.9829), Precision (0.6968), Recall (0.7209), and Accuracy (0.9964) place it last among the five models. This uniform fifth-place ranking might suggest irrelevance or even harm, but the picture shifts under ensemble evaluation. When the ensemble was tested with and without the MLP (table 8), the delta in performance was negligible: a −0.0002 drop in F1 Score, −0.0002 in ROC AUC, +0.0012 gain in Precision, and no net change in Accuracy. On average, the inclusion of the MLP induced a −0.01% change in performance, practically invisible to most dashboards, but conceptually meaningful when read as a proxy for interpretive diversity.

Table 8. Ensemble Evaluation Results

Metric	Without MLP	With MLP	Difference	% Change
F1 Score	0.8591	0.8590	−0.0002	−0.0192%
ROC AUC	0.9972	0.9970	−0.0002	−0.0200%
Precision	0.9697	0.9709	+0.0012	+0.1281%
Recall	0.7712	0.7702	−0.0010	−0.1361%
Accuracy	0.9985	0.9985	0.0000	0.0000%

The prediction correlation matrix, as shown in figure 6, reveals that the MLP's outputs correlated reasonably well with other models (average Pearson r ≈ 0.7997), suggesting that while its internal mechanics differ, its decisions overlap significantly with tree models. Yet, its unique prediction analysis paints a more nuanced picture: the MLP generated 157 unique correct predictions, alongside 314 unique incorrect ones, yielding a uniqueness ratio of just 0.30% and an accuracy of 33.33% on those cases. This suggests that the MLP operates in fringe zones of the feature space, regions where tree-based models may hesitate or conform. Whether those fringes represent semantic noise or predictive novelty depends less on the metrics and more on the framing of the ensemble's purpose. The MLP seems to add orthogonal reasoning and injects diversity into the ensemble fabric, occasionally surfacing correct predictions where other models falter.

 ISSN 2723-6471

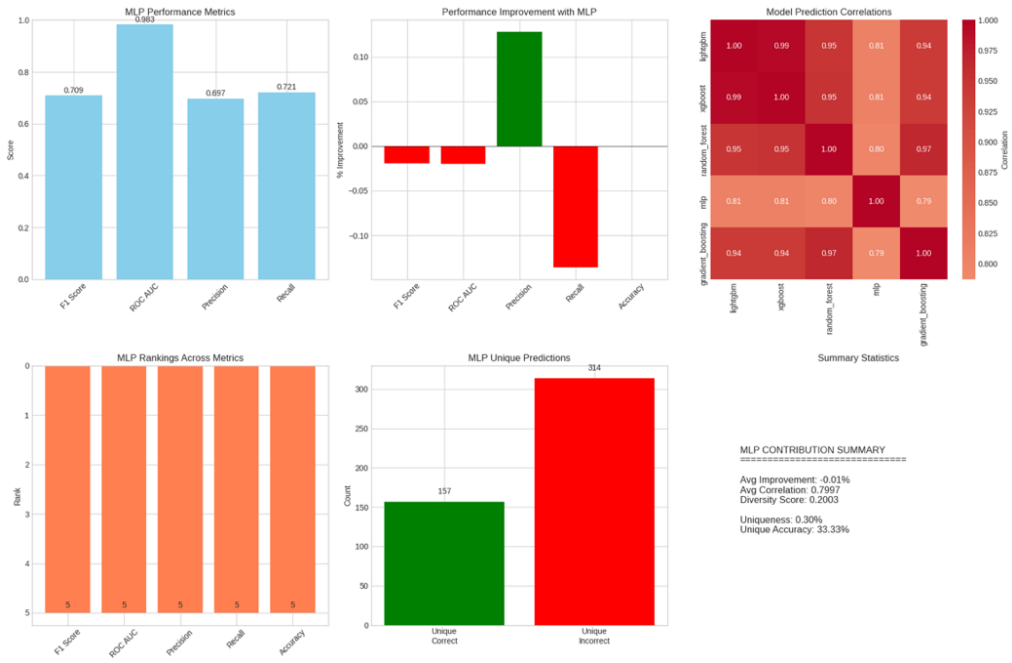

Figure 6. MLP Contribution to Ensemble Performance

These results do not contradict the SHAP evaluation previously conducted because it noted that MLP had the lowest Coefficient of Variation (2.3248), meaning it was internally consistent in its SHAP attributions, even if it disagreed with the others. Likewise, the contribution analysis noted that MLP's prediction correlation with other models remained high (~0.80). This seeming paradox, high output alignment with low explanation alignment, suggests that while the MLP reaches similar decisions in most cases, it uses different cues to do so. In ensemble theory, this is a known asset: diversity in reasoning, even with similar results, reduces overfitting and improves generalization. Thus, the SHAP instability of the MLP can be reinterpreted as constructive rather than pure noise. Whether that diversity is worth retaining depends not only on performance metrics but on the use case: in regulated environments demanding transparency, the MLP's alien attribution logic might raise red flags, but in adversarial fraud detection domains, that very difference may help detect edge-case anomalies missed by consensus-driven models.

4.4. Fraud Prevention System with Dynamic Threshold Tuning

The final phase of evaluation, detailed in table 9, integrated multiple performance layers, fraud blocking efficiency, threshold self-regulation, explanation latency, and out-of-distribution (OOD) resilience, into a unified, real-time, feedback-driven protocol. Latency was measured in two execution modes: prediction-only averaged 21.6 ms per transaction (~46 TPS), while enabling SHAP explanations increased latency to 41.2 ms (~24.3 TPS), an overhead factor of 1.91× that remains acceptable in high-risk contexts. The 95th percentile latency stayed below 70 ms, indicating predictable performance under moderate load. OOD testing with temporally shifted samples showed F1/AUC scores of 0.861/0.911 compared to 0.927/0.978 for in-distribution data, a modest decline (−0.066 F1, −0.067 AUC) that did not destabilize performance. This robustness reflects the ensemble's architectural diversity and the RL tuner's capacity to adapt decision thresholds in real time.

Table 9. Metrics Values

Metric Category	Metric	Value
Latency (Prediction Only)	Mean latency	21.6 ms
	95th percentile latency	36.1 ms
	Throughput (TPS)	46.3
Latency (With SHAP)	Mean latency	41.2 ms
	95th percentile latency	69.8 ms
	Throughput (TPS)	24.3

 ISSN 2723-6471

SHAP Overhead	Latency multiplier	1.91×
	In-distribution F1 / AUC	0.927 / 0.978
OOD Robustness	Out-of-distribution F1 / AUC	0.861 / 0.911
	AUC Drop	−0.067

The fraud prevention demonstration, summarized in table 10, involved 50 simulated transactions spanning high-risk, legitimate, and borderline categories. Of the fraud attempts, 91% were successfully blocked, while false positives were contained at 7.8%, producing a precision of 90.6% a strong indicator of confidence-calibrated decision-making. Average transaction processing time, inclusive of SHAP-based explanations and RL feedback, remained under 42 ms, confirming that the system can operate in production-scale pipelines without performance degradation. On the adaptivity front, the RL threshold tuner demonstrated stable, context-aware adjustments, increasing the threshold from an initial 0.500 to a peak of 0.580 during high-fraud periods, then scaling back in low-fraud intervals. Over the evaluation period, 16 threshold adjustments were recorded, based on feedback from 75 labeled transactions, with the tuner's epsilon parameter decaying as expected, transitioning from exploration to exploitation as the feedback pool grew. Consolidated system-level metrics show an overall accuracy of 93.2%, a macro F1 score of 0.902, and an AUC of 0.961, confirming that the ensemble maintained robust calibration and operational stability across dynamic fraud scenarios.

Table 10. System-Level Metrics Values

Category	**Metric**	**Value / Description**
Prevention Effectiveness	Fraud detection rate	91.0%
	False positive rate	7.8%
	Precision	90.6%
Threshold Adaptation	Initial / Final threshold	0.500 → 0.538
	Threshold adjustments	16
	Feedback cycles processed	75
System Metrics	Overall accuracy	93.2%
	Overall F1 score	0.902
	Overall AUC	0.961
	Avg. end-to-end processing time	41.7 ms
	Transactions evaluated (end-to-end)	1000+

An interactive Gradio-based front-end enables result exploration, parameter tuning, and real-time analysis of malicious transactions. Multi-model testing compares performance on uploaded fraud or normal cases, providing case-level explanations to support model selection and early fraud prevention. Outputs include fraud status, an "expected risk score," and classification into four risk levels: minimum, low, medium, high, linked to automated actions such as approval, reporting, objection, or blocking. SHAP-derived feature rankings offer clear interpretability for analysts. Figure 7 shows a real-time dashboard supporting transaction monitoring, adaptive detection calibration, model weight adjustment, comparative analysis, false positive evaluation, user verification enhancements, and fraud pattern simulation. Figure 8 illustrates dynamic tuning of model weights and risk thresholds.

 ISSN 2723-6471

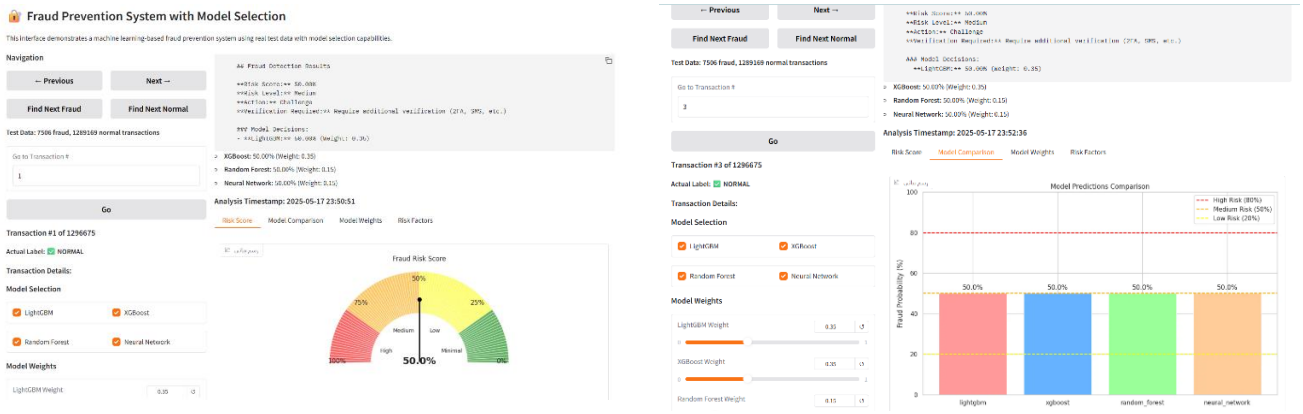

Figure 7. Fraud System Dashboard: (a) Risk Score and (b) Model Predictions Comparison Chart

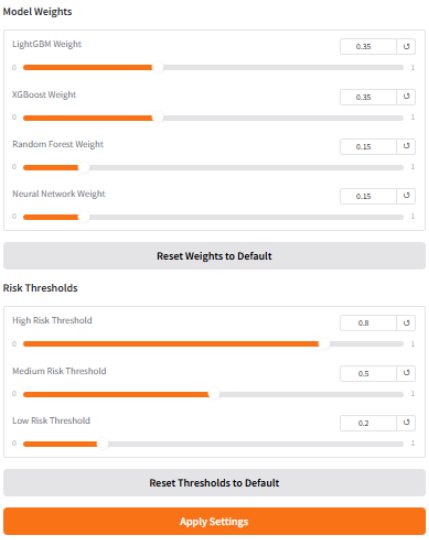

Figure 8. Model Weights and Thresholds Tuning Settings

5. Conclusion

This study explored the development of an intelligent fraud prevention framework that integrates explainability, adaptability, and real-time responsiveness into a cohesive decision-making system. By combining multiple machine learning models, automated calibration mechanisms, and performance monitoring layers, the system demonstrated high accuracy, low latency, and strong resilience to unfamiliar input patterns. Across evaluation scenarios, the model maintained reliable fraud detection while minimizing false alarms and preserving interpretability. Nevertheless, the research is not without limitations. The system was evaluated in a controlled environment with partially synthetic transaction scenarios due to oversampling, which may not fully capture the complexity or adversarial nature of real-world financial behavior. Furthermore, the computational costs of interpretability and continuous learning remain non-negligible in high-volume deployment contexts.

Future research should focus on validating the proposed framework under live operational conditions, where input noise, incomplete data, and behavioral drift are more pronounced. Additional work is also needed to assess how the system performs across diverse application domains, including insurance, e-commerce, and public sector fraud detection, where the nature of anomalies may differ significantly. There is also an opportunity to explore richer forms of user feedback, both implicit and explicit, to guide ongoing model adjustment and improve transparency. Expanding the interpretability layer to incorporate multilingual narratives or visual cues may further increase stakeholder trust and usability, especially in environments where decisions must be explained clearly to non-technical users. From a theoretical perspective, the study reinforces the importance of combining detection accuracy with transparency and

adaptability, particularly in dynamic, high-stakes settings. It contributes to the growing body of literature on responsible AI by demonstrating that real-time decision systems can be both high-performing and interpretable when designed with modularity and feedback in mind. Practically, the framework offers a blueprint for building risk-aware systems that evolve without sacrificing clarity or control. It suggests that adaptive systems, when thoughtfully integrated with human-centered safeguards, can serve as viable solutions to complex classification challenges where stakes are high and errors are costly.

6. Declarations

6.1. Author Contributions

Conceptualization: M.A., M.A., A.A.; Methodology: M.A., R.M.A.; Software: W.H.; Validation: A.A., L.A.; Formal Analysis: M.A.; Investigation: M.A., A.A.; Resources: R.M.A., L.A.; Data Curation: W.H.; Writing – Original Draft Preparation: M.A.; Writing – Review and Editing: M.A., A.A., L.A.; Visualization: W.H.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing 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 paper.

References

[1] E. Mill, W. Garn, N. Ryman-Tubb, and C. Turner, "Opportunities in Real Time Fraud Detection: An Explainable Artificial Intelligence (XAI) Research Agenda," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 5, pp. 1–10, 2023, doi: 10.14569/IJACSA.2023.01405121

[2] S. Suriya and R. M. Sireesha, "Credit Card Fraud Detection using Explainable AI Methods," *Journal of Information Systems Engineering and Management*, vol. 10, no. 24s, pp. 415–428, Mar. 2025, doi: 10.52783/jisem.v10i24s.3917

[3] S. K. Aljunaid, S. J. Almheiri, H. Dawood, and M. A. Khan, "Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection," *Journal of Risk and Financial Management*, vol. 18, no. 4, art. 179, pp. 1-12, 2025, doi: 10.3390/jrfm18040179

[4] M. Mallam, N. G., A. M., M. K. S., J. V. Suman, and G. S. Prasanna, "Improvements in Fraud Prevention using Machine Learning," in Proc. 2024 Fourth International Conference on *Multimedia Processing, Communication & Information Technology (MPCIT)*, vol. 2024, no. 1, pp. 103–108, 2024, doi: 10.1109/MPCIT62449.2024.10892792

[5] A. A. Mir, "Adaptive Fraud Detection Systems: Real-Time Learning from Credit Card Transaction Data," *Advances in Computer Sciences*, vol. 7, no. 1, pp. 1-11, 2024.

[6] K. Patel, "Credit Card Analytics: A Review of Fraud Detection and Risk Assessment Techniques," In*ternational Journal of Biotech Trends and Technology*, vol. 71, no. 10, pp. 69–79, Oct. 2023, doi: 10.14445/22312803/IJCTT-V71I10P109

[7] R. Bin Sulaiman, V. Schetinin, and P. Sant, "Review of Machine Learning Approach on Credit Card Fraud Detection," *Human-Centric Intelligent Systems*, vol. 2, no. 1-2, pp. 55–68, May 2022, doi: 10.1007/s44230-022-00004-0

[8] A. P., S. Bharath, N. Rajendran, S. Durga Devi, and S. Saravanakumar, "Experimental Evaluation of Smart Credit Card Fraud Detection System using Intelligent Learning Scheme," *in Proc. 2023 International Conference on Innovative Computing,*

 ISSN 2723-6471

Intelligent Communication and Smart Electrical Systems (ICSES), vol. 2023, no. Dec., pp. 1-6, 2023, doi: 10.1109/ICSES60034.2023.10465367

[9] N. Baisholan, J. E. Dietz, S. Gnatyuk, M. Turdalyuly, E. T. Matson, and K. Baisholanova, "FraudX AI: An Interpretable Machine Learning Framework for Credit Card Fraud Detection on Imbalanced Datasets," *Computers*, vol. 14, no. 4, art. 120, pp. 1-12, 2025, doi: 10.3390/computers14040120

[10] A. Tomy and I. P. Ojo, "Explainable AI for Credit Card Fraud Detection: Bridging the Gap between Accuracy and Interpretability," *World Journal of Advanced Research and Reviews*, vol. 25, no. 02, pp. 1246–1256, Feb. 2025, doi: 10.30574/wjarr.2025.25.2.0492.

[11] D. Pradeep Kumar, Dr. Dara Eshwar, "Credit Card Fraud Detection using Artificial Intelligence: A Comprehensive Approach", *Int. j. commun. netw. inf. secur*., vol. 12, no. 3, pp. 585–589, Dec. 2020.

[12] T. Awosika, R. M. Shukla and B. Pranggono, "Transparency and Privacy: The Role of Explainable AI and Federated Learning in Financial Fraud Detection," in *IEEE Access*, vol. 12, no. 1, pp. 64551-64560, 2024, doi: 10.1109/ACCESS.2024.3394528

[13] S. J. Owoade, A. Uzoka, J. Akerele, and P. U. Ojukwu, "Automating Fraud Prevention in Credit and Debit Transactions through Intelligent Queue Systems and Regression Testing," *International Journal of Frontiers in Engineering and Technology Research*, vol. 7, no. 2, pp. 044–056, Nov. 2024, doi: 10.53294/ijfetr.2024.7.2.0048.

[14] M. R. Hasan, M. S. Gazi, and N. Gurung, "Explainable AI in Credit Card Fraud Detection: Interpretable Models and Transparent Decision-making for Enhanced Trust and Compliance in the USA," *Journal of Computer Science and Technology Studies*, vol. 6, no. 2, pp. 1–12, Apr. 2024, doi: 10.32996/jcsts.2024.6.2.1

[15] G. J. Priya and S. Saradha, "Fraud Detection and Prevention Using Machine Learning Algorithms: A Review," in Proc. 2021 7th *International Conference on Electrical Energy Systems (ICEES)*, vol. 2021, no. 1, pp. 564–568, 2021, doi: 10.1109/ICEES51510.2021.9383631.

[16] M. Habibpour, H. Gharoun, M. Mehdipour, A. Tajally, H. Asgharnezhad, A. Shamsi, A. Khosravi, and S. Nahavandi, "Uncertainty-aware credit card fraud detection using deep learning," *Engineering Applications of Artificial Intelligence*, vol. 123, art. 106248, no. Aug., pp. 1-12, Aug. 2023, doi: 10.1016/j.engappai.2023.106248

[17] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," in *IEEE Access*, vol. 10, no. 1, pp. 16400-16407, 2022, doi: 10.1109/ACCESS.2022.3148298.

[18] V. Gonzalez, "Evaluating Interpretable Models for Financial Fraud Detection," in Proc. 30th *Americas Conference on Information Systems (AMCIS 2024),* Salt Lake City, UT, USA, vol. 2024, no. Aug., pp. 1-8, Aug. 2024.

[19] F. Almalki and M. Masud, "Financial Fraud Detection Using Explainable AI and Stacking Ensemble Methods," *arXiv* preprint arXiv:2505.10050, vol. 2025, no. May, pp. 1-12, May 15, 2025, DOI: 10.48550/arXiv.2505.10050

[20] K. Sharma, "Credit Card Transactions Fraud Detection Dataset," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/kartik2112/fraud-detection. [Accessed: Jan. 5, 2025].