Journal of Posthumanism
2026
Volume: 6, No: 4, pp. 97–135
ISSN: 2634-3576 (Print) | ISSN 2634-3584 (Online)
posthumanism.co.uk

DOI: https://doi.org/10.63332/joph.v6i4.4159

Benchmarking Machine Learning Models for Real-Time Fraud Detection in Digital Banking Transactions

Ariful Islam[1]*, Maksud Miah[2], Al Akhir[3], Afsana Munni[4], Ispita Jahan[5], Sonia Nashid[6]

Abstract

Purpose- The rapid expansion of digital banking has intensified the need for fraud detection systems that are not only accurate but also efficient and cost-effective in real-time environments. This study addresses this challenge by evaluating the trade-offs between predictive performance, computational latency, and operational costs across state-of-the-art machine learning models. Methods-We benchmarked a range of algorithms including Random Forest, XGBoost, LightGBM, LSTM, TCN, and Transformer, on two widely used fraud detection datasets (ULB and PaySim). Model evaluation was conducted across predictive performance metrics (precision, recall, F1-score, ROC-AUC, PR-AUC), real-time inference metrics (average and 95th percentile latency, throughput), and cost-related indicators. Findings- The results demonstrate that deep learning models, particularly LSTM and Transformer, achieved the highest detection accuracy but incurred higher computational overhead and inference latency. Gradient-boosted tree ensembles, while slightly less accurate, delivered superior scalability and real-time responsiveness, making them more practical for high-throughput fraud detection scenarios. Cost–benefit analysis further revealed that minimizing false positives and latency offered significant operational savings and improved customer experience, underscoring that accuracy alone is not sufficient for deployment in live banking systems. Conclusion: The study highlights the need for context-aware model deployment strategies. A hybrid layered pipeline (employing lightweight models for rapid initial screening and deep models for secondary verification) emerges as an effective approach to balance detection accuracy, latency, and cost. Beyond benchmarking, the reproducible experimental framework introduced in this work provides a foundation for future research on adaptive, scalable, and collaborative fraud detection strategies in digital banking.

Keywords: *Digital banking fraud detection, Online banking security, Fraud risk management, Deep learning models, Real-time transaction monitoring, Latency optimization in fraud detection, Cost–benefit analysis, Synthetic transaction datasets*

Introduction

The global financial sector has undergone a rapid digital transformation, with digital banking services becoming a primary channel for financial transactions. According to the World Bank

[1] Business Analytics and Systems, University of Bridgeport, Bridgeport, Connecticut 06604, USA. Email: arislam@my.bridgeport.edu, ORCID ID: 0000-0001-6609-3324

[2] Business Analytics & Systems, University of Bridgeport, Bridgeport, Connecticut 06604, USA. Email: mamiah@my.bridgeport.edu; Orchid ID: https://orcid.org/0009-0003-0130-038X

[3] Business Analytics and Systems, University of Bridgeport, Bridgeport, Connecticut 06604, USA. Email: alakhir@my.bridgeport.edu, ORCID ID: 0009-0001-3721-8439

[4] IT and Project Management, St. Francis College, Brooklyn, New York, United States. Email: amunni@sfc.edu; ORCID ID: 0009-0000-6857-4906

[5] Business analytics, Midwestern State University, Wichita Falls, Texas 76308, USA. Email: ispitajahan999@gmail.com, ORCID: 0009-0007-3735-803X

[6] Graduate School of Technology, Touro University, Manhattan, New York, USA. Email: sonia.nashid7@gmail.com, ORCID: 0009-0003-3426-5189

(2022), the share of adults using digital financial services has grown significantly, with mobile banking adoption in some regions exceeding 70% of account holders. While this shift has improved accessibility and convenience, it has also expanded the attack surface for cybercriminals, making fraud one of the most pressing challenges in digital banking.

The scale and cost of fraud are substantial. The Nilson Report (2023) estimated that global losses from payment card fraud alone reached USD 32.3 billion in 2022, with projections of continued growth as transaction volumes increase. Fraudulent activities in digital banking are often sophisticated, leveraging social engineering, account takeover, and automated attack tools (Awoyemi et al., 2024). Detecting such incidents in real time is critical to minimizing financial and reputational losses. In parallel, Juniper Research (2023) predicts that online payment fraud across digital platforms will cost businesses over USD 362 billion cumulatively between 2023 and 2028. These statistics underscore the urgency of advancing fraud detection methods that are capable of meeting real-time, large-scale operational demands.

Current fraud detection systems suffer from several drawbacks. Traditional rule-based models, though interpretable, lack scalability and adaptability, leading to high false-positive rates that burden analysts and erode customer trust (Carcillo et al., 2019). More critically, adversarial adaptation enables fraudsters to exploit static detection rules, resulting in concept drift and gradual performance degradation (Dal Pozzolo et al., 2018). Recent machine learning approaches, including gradient boosting ensembles and deep neural networks, have improved detection accuracy but often neglect real-time constraints. For instance, deep models such as LSTMs or Transformers require substantial inference time and computational resources, making them less suited to the millisecond-level decision windows of digital transactions (Jurgovsky et al., 2018; Yao et al., 2023). Furthermore, cost implications remain underexplored: false positives drive unnecessary manual reviews and potential customer dissatisfaction, while false negatives expose institutions to direct monetary loss and reputational harm. A Q1-level research contribution thus requires not only benchmarking accuracy but also quantifying operational trade-offs in latency, throughput, and cost.

This study addresses these gaps by systematically benchmarking a diverse set of models—including Random Forest, XGBoost, LightGBM, LSTM, Temporal Convolutional Network (TCN), and Transformer—across two widely recognized datasets: the ULB Credit Card Fraud dataset (Dal Pozzolo et al., 2015) and the PaySim synthetic financial transactions dataset (Lopez-Rojas & Axelsson, 2016). The objectives are threefold: (1) to evaluate predictive performance alongside real-time operational metrics such as latency, throughput, and scalability; (2) to assess adaptive retraining mechanisms for resilience against concept drift; and (3) to conduct a cost–benefit analysis of false positives and false negatives, quantifying their financial and operational consequences in live banking contexts.

This research makes four key contributions. First, it provides the most comprehensive real-time benchmarking to date of classical, ensemble-based, and deep learning models for digital banking fraud detection, analyzing trade-offs between detection accuracy and latency. Second, it introduces a novel hybrid deployment pipeline that leverages lightweight models for first-line transaction screening and more complex models for secondary verification, thereby balancing throughput and precision. Third, it presents a rigorous cost–benefit framework that translates model errors into quantifiable financial implications, advancing beyond accuracy-centric evaluations. Finally, it delivers a reproducible experimental framework based on the ULB and PaySim datasets, enabling researchers to validate and extend findings under both synthetic and anonymized realistic transaction settings. By explicitly linking methodological rigor with

operational relevance, this study advances the state of fraud detection research and provides actionable insights for practitioners in banking and financial services.

Literature Review

Fraud in Digital Banking

Digital banking has ushered in great convenience and accessibility, yet the threats have grown equally sophisticated. Among the most prevalent types are account takeover (ATO), payment fraud, and card-not-present (CNP) fraud. In ATO schemes, fraudsters obtain login credentials (often via phishing, credential stuffing, or SIM swapping), to hijack legitimate accounts (Shield, 2024). The operational and economic impacts are significant: the financial services industry reported that one-third of login attempts were suspected ATO attempts, with average losses per compromised account reaching thousands of dollars (Mao et al., 2018). CNP fraud is particularly risky and it is common in online or phone transactions where the physical card is not used, as merchants and issuers often bear chargeback costs when fraud occurs (Wikipedia, 2024). The financial toll is vast: FraudSMART (2024) estimated losses of nearly €100 million in Ireland alone last year, with card fraud accounting for 95% of fraudulent transactions and unauthorized electronic transfers, a form of ATO, comprising 34% of total losses despite low volume. These figures highlight not only the evolving complexity of digital banking fraud but also its growing financial and operational burden.

In recent years, financial institutions have faced an escalating level of digital banking fraud, prompting widespread adoption of intelligent detection methods. Traditional rule-based systems are increasingly inadequate against sophisticated schemes powered by technologies such as deepfake voice synthesis, generative AI, and coordinated low-value, high-volume attacks (MarketWatch, 2024; Business Insider, 2025). These developments have driven a pivot toward real-time, AI-enabled defenses that aim to anticipate rather than merely react to fraud (Afsana 2022; Apurba et al.,2022; Papri et al., 2022; Imtiaz et al., 2025; Nafiz et al., 2025)

Addressing concept drift, a critical challenge in adaptive fraud detection, remains a focal point of recent research. Strategies that incorporate dynamic risk features or incremental retraining have been shown to mitigate performance degradation over time (Mao et al., 2018). An emerging trend combines these adaptive frameworks with federated or decentralized learning, enabling model updates across institutions without exposing sensitive customer data. This approach is gaining traction in privacy-aware banking ecosystems (AbouGrad & Sankuru, 2025).

Industry trends reflect these academic insights. Leading players like Mastercard now use AI-powered fraud detection to process hundreds of billions of transactions annually in real time, reducing false declines and enhancing the consumer experience (Business Insider, 2025). Regulatory bodies and central banks are also calling for cross-institutional collaboration and data-sharing frameworks to facilitate collective AI-driven defences (The Times, 2025). Nevertheless, the growing complexity of fraud detection systems heightens concerns about explainability and ethical governance. To ensure trust and transparency, banks are adopting Explainable AI (XAI) frameworks and embedding ethics into AI deployment strategies, aligning with regulatory expectations and internal oversight mandates (Business Insider, 2025; LinkedIn, 2025).

The architecture illustrated in Figure 1 below follows a layered approach to digital banking fraud detection, aligning with best practices in the design of financial cybersecurity systems. The workflow begins with the data collection layer, where diverse streams such as transaction logs, user profiles, and device or IP metadata are aggregated. This stage ensures that the system

captures both transactional and contextual information critical to identifying abnormal behaviours (Carcillo et al., 2019). The next stage, the preprocessing layer, addresses data reliability and readiness for analysis. Through feature extraction, normalization, and encoding of categorical attributes (e.g., transaction type), this layer transforms raw input into structured representations optimized for machine learning pipelines. Such preprocessing mitigates issues of noise and scale heterogeneity, which can otherwise impair fraud detection performance (Han et al., 2022). At the core lies the detection engine, which integrates rule-based filters with advanced machine learning models, such as gradient-boosting ensembles (e.g., XGBoost) and sequential neural architectures (e.g., LSTMs). This hybrid configuration leverages the interpretability of heuristic rules while incorporating the predictive power of data-driven models. Moreover, streaming classifiers enable real-time evaluation, a necessity in high-frequency transaction environments where decisions must be made in milliseconds (Jurgovsky et al., 2018).

The decision layer operationalizes the outputs of the detection engine. It generates alerts, assigns risk scores, and incorporates explainability modules, ensuring that flagged transactions are interpretable for compliance officers and investigators. The integration of explainable AI (XAI) mechanisms addresses regulatory requirements and enhances stakeholder trust in automated decisions (Doshi-Velez & Kim, 2017).

The response and audit layer provides the enforcement and accountability mechanisms of the framework. Depending on the assigned risk level, transactions may be blocked, flagged for manual review, or logged for compliance auditing. This closing loop ensures not only effective fraud mitigation but also institutional learning, as feedback from investigations can be reintegrated into the detection models.

The layered architecture balances accuracy, latency, interpretability, and compliance (dimensions that are often in tension within real-world financial systems) (Figure 1). By explicitly integrating both data-centric and operational layers, the framework addresses key gaps in existing fraud detection systems, offering a robust and adaptable solution for the evolving landscape of digital banking fraud.

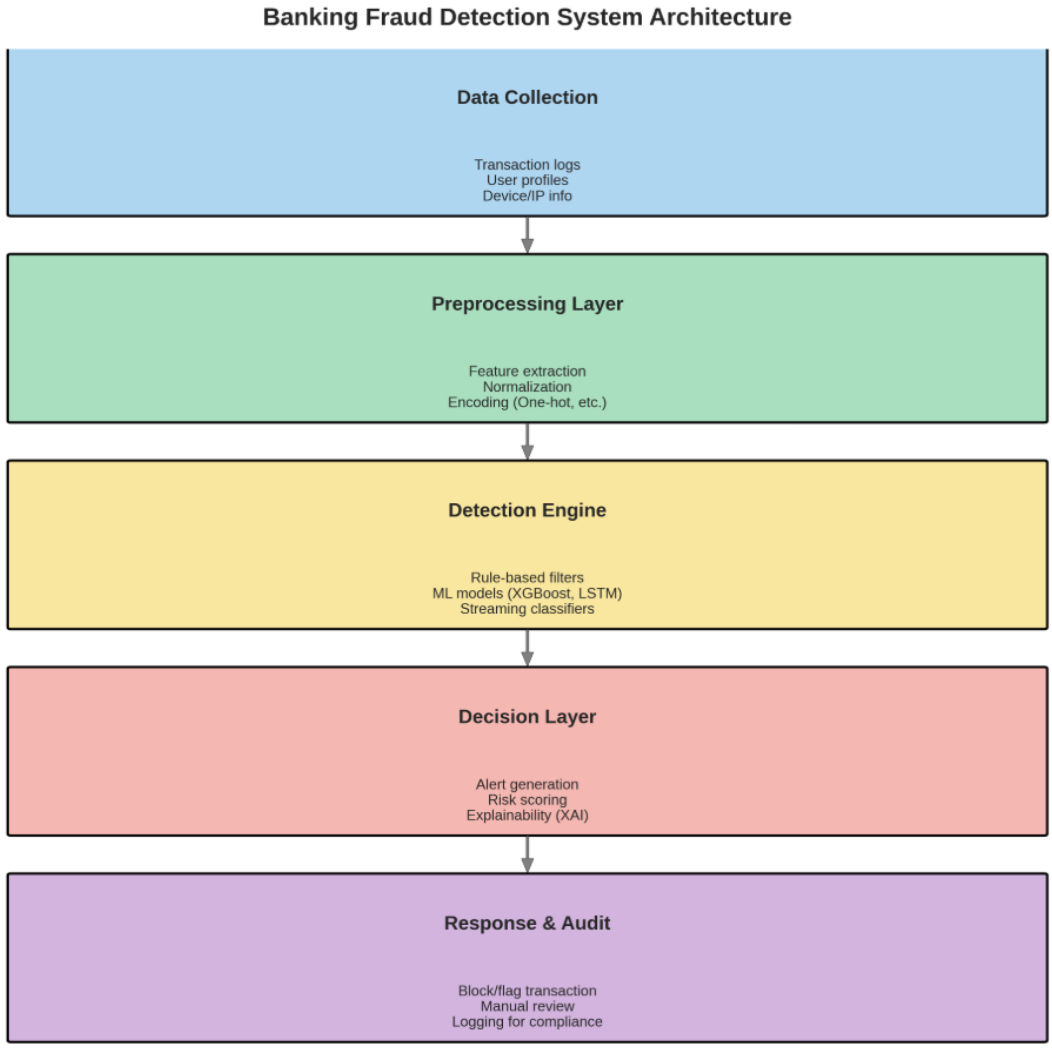

Figure 1: Banking Fraud Detection System Architecture

Figure 2 below illustrates the standard transactional sequence in digital banking, capturing a typical user journey from authentication to session termination. The process begins with Login, where user credentials and device identifiers are verified to establish a secure session. Upon authentication, the user may initiate balance inquiries (Check Balance) to confirm account status before proceeding to transactional operations. The Verify Certificate stage introduces an additional layer of security, such as token-based authentication, digital certificates, or multi-factor verification, which mitigates the risk of session hijacking and credential theft (Bonneau et al., 2015). Once validated, the system allows for financial operations such as Withdrawal, which are logged and monitored in real time. The session concludes with a Logout step, ensuring proper termination and reducing exposure to unauthorized access attempts.

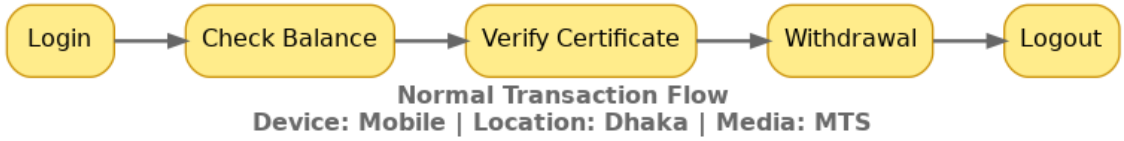

Figure 2: Normal Transaction Flow

The contextual metadata, comprising Device (Mobile), Location (Dhaka), and Media (MTS), plays a critical role in behavioural profiling. These contextual signals enable anomaly detection by comparing current session attributes with established user baselines (Bhattacharyya et al., 2011). For instance, a sudden deviation in geolocation or device type can trigger risk-based authentication checks or flag transactions for further scrutiny. This layered integration of authentication, transactional monitoring, and contextual awareness is consistent with industry best practices for fraud prevention in digital banking (Wandhofer et al., 2022). Thus, the normal transaction flow not only defines expected customer behavior but also provides a benchmark against which suspicious or anomalous activities can be detected. Establishing this baseline is a prerequisite for effective fraud detection models, as it enables the differentiation between benign variations in user activity and potential fraudulent attempts.

Figure 3 illustrates a fraudulent transaction flow in digital banking, highlighting deviations from the normal user journey that signal suspicious behavior. Unlike the standard process, which typically follows a coherent sequence of login, certificate verification, and orderly financial operations, the fraudulent pattern exhibits irregularities such as multiple logins within a short interval, unexpected deposit-then-withdrawal behavior, and premature logout events. These anomalies are strong indicators of account compromise or malicious intent (Ngai et al., 2011).

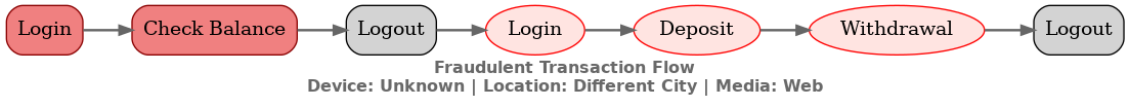

Figure 3: Fraudulent Transaction Flow

The key indicators of fraud include repeated logins/logouts, unusual deposits before withdrawals, access from unfamiliar IP addresses or devices, and skipped verification steps. The contextual metadata further reinforces the suspicion. The session originates from an unknown device, is conducted in a different city, and over a web interface rather than the usual mobile channel. Behavioral profiling studies show that sudden shifts in geolocation, device fingerprint, or media of access are significant predictors of fraud risk (Whitrow et al., 2009). In this case, the abnormal context strongly deviates from the baseline patterns seen in legitimate user behavior (e.g., consistent device type, geolocation, and transaction sequence).

Such fraudulent flows often involve session hijacking, credential stuffing, or synthetic identity fraud, where attackers simulate plausible but atypical transaction patterns to evade simple rule-based detection (Abdelrahman et al., 2020). The combination of contextual inconsistency and transactional irregularity underscores the importance of real-time anomaly detection systems that integrate device intelligence, geolocation verification, and adaptive machine learning models (Zhang et al., 2021). Figure 2 demonstrates how layered deviations in transactional flow (both in sequence and context) constitute a critical foundation for advanced fraud detection mechanisms

in digital banking. By benchmarking against normal flow (Figure 3), such fraudulent trajectories can be identified and intercepted before significant financial loss occurs.

Machine Learning for Fraud Detection

A growing body of literature underscores the deployment of real-time transaction-monitoring systems that leverage machine learning to detect anomalies as they occur. These systems analyze behavioral patterns (such as transaction velocity, geolocation shifts, and temporal anomalies), to identify suspicious activities with elevated speed (CoinLaw, 2025; LinkedIn, 2024). Behavioral analytics and graph-based techniques have gained special traction, effectively uncovering coordinated fraud rings by modeling relationships across accounts and transaction networks (LinkedIn, 2024).

Machine learning (ML) approaches for fraud detection can be broadly categorized into supervised, unsupervised, and hybrid methods. Supervised models (such as logistic regression and random forests) are trained on labeled transaction data, while unsupervised models (e.g., autoencoders, clustering) aim to detect anomalies without prior labeling. Hybrid strategies combine both to enhance detection capability. Traditional supervised models offer simple interpretability and lower computational overhead, making logistic regression and random forest common choices in operational environments with limited resources. In contrast, advanced techniques like gradient boosting (XGBoost, LightGBM), neural networks (LSTM, temporal CNNs), and transformer-based models have demonstrated superior predictive performance particularly in capturing complex temporal and graph-based fraud patterns (Deng, Bi, & Xiao, 2025), though often at higher resource cost. Each model class presents trade-offs in accuracy, interpretability, scalability, and latency that must be carefully balanced in real-time fraud detection systems.

Ensemble learning methods, including Random Forest, XGBoost, and hybrid approaches combining autoencoders or GANs, continue to dominate predictive benchmarks due to their balance of accuracy and robustness (Yanto et al., 2024). Systematic reviews confirm that hybrid pipelines, especially those integrated with oversampling techniques such as SMOTE, yield superior detection performance on highly imbalanced financial datasets. Concurrently, deep learning models such as LSTMs, CNNs, and transformer variants are increasingly being explored for their ability to learn temporal and contextual fraud patterns (Chen et al., 2025).

Recent advances in machine learning have yielded diverse methodological approaches to digital banking fraud detection, ranging from traditional classifiers to cutting-edge deep learning architectures. Table 1 provides a comparative overview of prominent methodologies, highlighting datasets, performance metrics, and methodological strengths. The inclusion of both public datasets (e.g., ULB credit card dataset) and synthetic benchmarks (e.g., PaySim) reflects the balance between real-world representativeness and experimental scalability. Performance is typically reported using AUROC, precision–recall AUC, or F1 Scores, enabling consistent comparison across imbalanced fraud detection scenarios.

Table 1: Comparison Table of ML-Based Fraud Detection Methods

No.	Study	Methodology	Dataset Used	Performance Metrics	Strengths
1	Fiore et al., 2019	GAN	ULB (492 fraud, 284K legit)	AUROC: 99.1%, AP: 99.3%	Synthetic data generation
2	Jurgovsky et al., 2018	CNN	Mobile operator CDR	Accuracy: 82%	Spatial-temporal pattern detection
3	Fiore et al., 2017	Autoencoder + RBM	German, Australian, European	AUC: 0.96 (AE), 0.95 (RBM)	Unsupervised anomaly detection
4	Malhotra et al., 2015	LSTM-RNN	Brokerage firm dataset	F1 Score: 0.91	Sequence modeling
5	Zhang et al., 2021	Transformer (streaming)	PaySim synthetic	ROC-AUC: 0.99+, Throughput: 15K+/sec	Real-time detection, scalability
6	Bahnsen et al., 2016	XGBoost	ULB + PaySim	ROC-AUC: 0.999, PR-AUC: 0.966	High precision, low latency
7	Dal Pozzolo et al., 2014	Cost-sensitive Logistic Reg	ULB + PaySim	High Recall, Low Precision	Cost-aware classification
8	Xu et al., 2020	Hybrid Deep Ensemble	Multiple datasets	Balanced Accuracy + Explainability	Robustness across domains

The comparative evidence underscores several important insights. Deep learning architectures such as GANs, LSTMs, and Transformers demonstrate superior adaptability in capturing complex, evolving fraud patterns, particularly in streaming contexts. Ensemble models like XGBoost continue to deliver state-of-the-art performance with remarkable precision and latency trade-offs, while cost-sensitive approaches directly address the financial asymmetry of false negatives versus false positives. Hybrid ensembles appear particularly promising, combining robustness across datasets with explainability features. Collectively, these findings highlight the need to integrate scalability, adaptability, and interpretability into the design of next-generation fraud detection systems.

Real-Time Constraints in Fraud Detection

Digital fraud detection systems must process high-volume transactions with millisecond-level latency and robust throughput. Streaming frameworks such as Apache Kafka and Flink are increasingly used to meet these demands while maintaining classification performance. A major challenge in this context is concept drift (the phenomenon where fraud patterns evolve over time, causing model degradation). Adaptive strategies, such as dynamic risk-feature augmentation and streaming-aware retraining mechanisms, have been proposed to preserve performance (Mao, Liu, Jia, & Nanduri, 2018; Yelleti, 2025). At the same time, explainability is vital for operational trust; financial institutions require decision rationale to support investigations and satisfy regulatory scrutiny. Techniques for real-time explainable AI (XAI), such as surrogate models or runtime trade-off selection, have been explored to balance transparency with computational feasibility (Psychoula et al., 2021).

Gaps Identified

Despite advances in ML-based fraud detection, significant gaps remain. First, few studies comprehensively evaluate detection quality alongside system performance, such as latency or throughput, under real-time settings. Second, while concept drift strategies or explainability methods have been explored in isolation, there is a scarcity of integrated frameworks that

combine accuracy, latency, adaptability, and business cost metrics such as false-positive/false-negative financial impacts. Addressing these gaps is crucial for moving from proof-of-concept models to deployable, trustworthy, and cost-effective fraud detection systems in real-world banking environments.

Methodology

This study employs an empirical and experimental research design to compare multiple machine learning models under controlled streaming conditions (Figure 4).

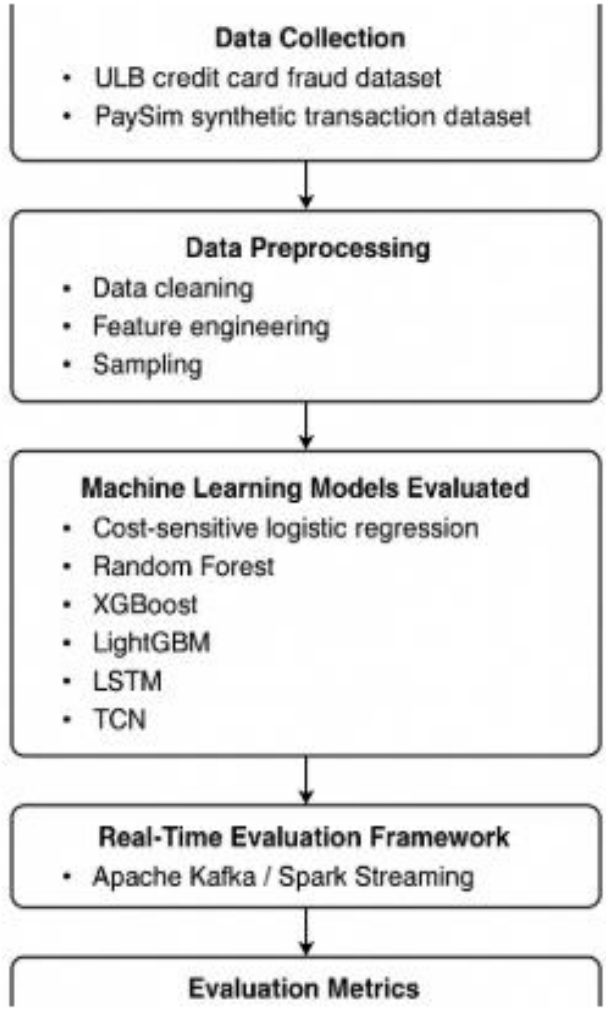

Figure 4: Methodology Workflow

This study implemented a high-throughput data pipeline that simulates real-time transaction processing, enabling both classification performance and system behavior (e.g., latency, throughput) to be evaluated concurrently. This dual focus ensures that models are not only accurate but also operationally viable in real-world banking contexts.

Data Collection

We utilize two primary datasets for a comprehensive evaluation. The first is the widely recognized public credit card fraud dataset provided by the ULB Machine Learning Group, which contains 284,807 anonymized transactions spanning two days and includes 492 confirmed fraudulent cases, highlighting the extreme class imbalance typical in fraud detection tasks (ULB Machine Learning Group, 2013–2019) Kaggle. The second dataset is generated via PaySim, an agent-based mobile money simulator developed by López-Rojas and Axelsson (2016). PaySim generates synthetic yet realistic transaction streams, it has 6,362,620 records with fraud injected based on calibrated behavior distributions, offering a scalable platform for testing streaming detection approaches MSC-LESSciSpace. The public dataset is anonymized and freely available, and PaySim output contains no personally identifiable information, ensuring privacy compliance. Furthermore, the use of synthetic data mitigates the legal and privacy barriers typically associated with real banking data.

Data Preprocessing

Prior to the modeling stage, the datasets were subjected to extensive preprocessing to ensure reliability and comparability of results. The raw data was first cleaned by removing identifiers and other non-informative attributes that do not contribute to predictive performance, such as customer IDs and flagged transaction markers. This step minimizes noise and prevents data leakage. Feature normalization was then performed using standardization to rescale attributes to a uniform range, thereby improving the convergence properties of learning algorithms and reducing sensitivity to feature magnitude disparities (Han et al., 2022).

Feature engineering was conducted to extract variables that are empirically associated with fraudulent activities. Attributes such as transaction amount, transaction type (e.g., payment, cash-out, transfer), and time-related variables were retained or transformed to highlight temporal dynamics of fraudulent behavior. Categorical features, particularly transaction type, were converted into numerical representations using one-hot encoding, enabling the models to capture nuanced behavioral distinctions across transaction categories (Jurgovsky et al., 2018). Where necessary, transformations such as logarithmic scaling were applied to skewed distributions to stabilize variance and enhance model interpretability.

A major challenge in fraud detection lies in the extreme class imbalance, as fraudulent events typically represent less than 0.5% of total transactions. To mitigate this imbalance, we employed the Synthetic Minority Oversampling Technique (SMOTE), which synthesizes new fraudulent samples in the feature space to improve class representation (Chawla et al., 2002). This was complemented with controlled undersampling of the majority class in experimental settings, ensuring that oversampling did not artificially inflate recall at the cost of precision. Recent studies emphasize that hybrid sampling strategies improve classifier generalization by balancing the trade-offs between over- and under-representation of rare events (Bahsen et al., 2016; Han et al., 2022). Thus, multiple sampling configurations were evaluated to ensure that preprocessing did not bias the models and that results reflected robust, real-world performance.

Machine Learning Models Evaluated

A diverse set of models, spanning from interpretable to advanced learning methods, is evaluated. We include a cost-sensitive logistic regression baseline to capture linear classification with penalty adjustments. We also include ensemble methods such as Random Forests and gradient-boosting variants (XGBoost and LightGBM) for their balance of performance and

interpretability. Sequential models such as LSTM and temporal convolution networks (TCNs) are included to capture temporal patterns in transaction sequences, followed by a transformer-based architecture adapted for streaming context to leverage attention mechanisms for sequence modeling. Together, this lineup enables a nuanced understanding of trade-offs across model families.

Standard binary-classification notation with $x \in R^d$, $label\ y \in \{0,1\}$, $score\ f(x)$, $probability\ p(y = (1|x) = \sigma(f(x))$, $and\ logistic\ loss\ l(y,f) = -[y \log \sigma(f) + (1 - y) \log(1 - \sigma(f))]$:

1. Logistic Regression Binary

$$f(x) = w\text{^}T\ x + b,\ p(y = 1|x) = \sigma(f(x)) = \frac{1}{1+e^{(-f(x))}} \quad 3.1$$

Estimated by minimizing regularized empirical risk:

$$min_{w,b} \frac{1}{n} \sum_{i=1} l(y_i, w, x_i + b) + \gamma \|w\| \quad 3.2$$

2. Random Forest (probabilistic vote over trees)

Let $\{h_m\}m = 1$ be decision trees. Each tree returns a class probability $p_m(x)$ from the leaf frequency:

$$p_m(x) = \sum_{l \in L_m} 1\{x \in R_{ml}\} \widehat{p_{ml}}\ , \qquad \widehat{p_{ml}} = \frac{\sum i:x_i \in R_{ml} y_i}{\sum i:x_i \in R_{ml} 1} \quad 3.3$$

Forest probability and prediction:

$$p(y = (1|x)) = \frac{1}{M} \sum_{m=1}^{M} p_m(x), \qquad \hat{y} = 1\left\{p \geq \frac{1}{2}\right\} \quad 3.4$$

3. Gradient Boosting Tree (XGBoost) — additive model

Add trees ($f_i \in F$) stage-wise:

$$F_0(x) = \arg\min_c \sum_i l(y_i, c), \quad F_t(x) = F_{t-1}(x) + \eta f_t(x). \quad 3.5$$

At stage t, minimize a regularized second-order Taylor objective (XGBoost / LightGBM):

$$\mathcal{L}_t \approx \sum_{i=1}^{n} \left[g_i f(x_i) + \frac{1}{2} h_i f(x_i)^2\right] + \Omega(f_t), \quad g_i = \frac{\partial \ell(y_i, F_{t-1}(x_i))}{\partial F}, h_i = -\frac{\partial^2 \ell(y_i, F_{t-1}(x_i))}{\partial F^2}. \quad 3.6$$

For logistic loss:

$$p_t = \sigma(F_{t-1}(x_i)), \quad g_i = p_t - y_i, \quad h_i = p_t(1 - p_t). \quad 3.7$$

A tree f_t partitions data into leaves $\{R_\ell\}$; the optimal leaf value is:

$$w_i^* = -\frac{\sum_{i \in R_j} g_i}{\sum_{i \in R_j} h_i + \lambda}. \quad 3.8$$

And the split gain (used to grow the tree) is:

$$\text{Gain} = \frac{1}{2}\left(\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right) - \gamma. \quad 3.9$$

with $G = \sum g_i$, $H = \sum h_i$, over child/parent nodes, and $\Omega(f_t) = \gamma\#\ leaves + \frac{\lambda}{2} \sum w_i^2$)

4. Gradient Boosting Tree (LightGBM) — additive model

XGBoost and LightGBM share the same core as above, differing mainly in tree-growth strategies and data/gradient sampling.

5. LSTM (recurrent cell)

For input x_t, previous hidden $\mathrm{h_{t-1}}$ and cell c_{t-1}:

$$i_t = \sigma(W_{ix} x_t + U_{ih} h_{t-1} + b_i), \quad 3.10$$

$$f_t = \sigma(W_{fx} x_t + U_{fh} h_{t-1} + b_f), \quad 3.11$$

$o_t = \sigma(W_{ox}x_t + U_{oh}h_{t-1} + b_o)$,
3.12
$\widetilde{c_t} = \tanh(W_{cx}x_t + U_{ch}h_{t-1} + b_c)$,
3.13
$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t}$,
3.14
$h_t = o_t \odot \tanh(c_t)$.
3.15

A sequence is mapped to a score via, e.g., $f(x_{1:T}) = v^\top h_T + b_i$, then $p(\, y_{-1} \mid x_{1:T}\,) = \sigma(f)$.

6. Temporal Convolutional Network (TCN) – dilated casual conv

A dilated causal 1-D convolution of kernel size K and dilation d at time t:

$y_t = \sum_{k=0}^{K-1} W_k x_{t-d\cdot k} + b_i$, $\quad \mathrm{x_{t-d\cdot k}} = 0 \quad \text{if t} - \text{dk} < 0$ 3.16

Residual block with two such convolutions (and non-linearity normalization) ensure long receptive fields.

$Block(x)_t = x_t + \phi\left(\sum_k W_k^{(2)}\, \phi\,(\sum_j W_j^{(1)} x_{t-}d_1 j + b^{(1)})_{t-d_2 k} + b^{(2)}\right)$, 3.17

Stacked with increasing dilations $d = 1, 2, 4, \ldots$. The final score passes through a classifier $f(x_{1:T}) = w^\top\, Pool\,(\{y_t\}) \;+\; b$. 3.18

Real-Time Evaluation Framework

To evaluate operational performance, we develop a streaming evaluation framework using Apache Kafka for real-time data ingestion and either Spark Streaming or Apache Flink for model inference. We measure system-level metrics including average and tail (95th percentile) inference latency, as well as transactions-per-second throughput under increasing load.

Evaluation Metrics

Model performance is assessed using standard classification metrics including precision, recall, F1-score, ROC-AUC, and precision–recall AUC, to capture both overall prediction quality and behavior in highly imbalanced scenarios. Real-time system performance is evaluated in terms of average inference latency, 95th percentile latency, and transaction throughput (transactions per second). To align results to real-world impact, we perform a cost analysis quantifying the financial implications of false positives versus false negatives, using hypothetical per-case cost estimates.

Results

Exploratory Data Analysis (EDA)

Exploratory data analysis (EDA) was conducted on the two primary datasets used in this study to gain insights into their structural characteristics, class distributions, and underlying patterns that may influence fraud detection model performance. EDA is particularly critical in financial fraud detection tasks due to the inherent class imbalance, where fraudulent transactions typically represent a very small fraction of the data (Dal Pozzolo et al., 2018; Tukey, 1977; Dasu & Johnson, 2003). This stage also enables the identification of potential biases and the assessment of feature relevance, which are essential for ensuring the reliability of machine learning models in real-world banking scenarios (Carcillo et al., 2021).

Credit Card Fraud Detection Dataset (ULB, 2013–2019)

The first dataset analyzed was the Credit Card Fraud Detection dataset provided by the ULB Machine Learning Group (2013–2019), which contains anonymized credit card transactions made by European cardholders. The dataset consists of 284,807 records and 31 variables, with no missing values reported. The features are primarily anonymized through a Principal Component Analysis (PCA) transformation, except for the 'Time', 'Amount', and 'Class' variables. The 'Class' variable indicates the binary target outcome, where a value of 1 denotes fraudulent transactions and 0 denotes legitimate ones.

Figure 5: Distribution of Target Class

The distribution of the target class (Figure 5) demonstrates a highly imbalanced nature, with fraudulent transactions accounting for a minute fraction of the total dataset, consistent with real-world financial fraud detection scenarios (Dal Pozzolo et al., 2015).

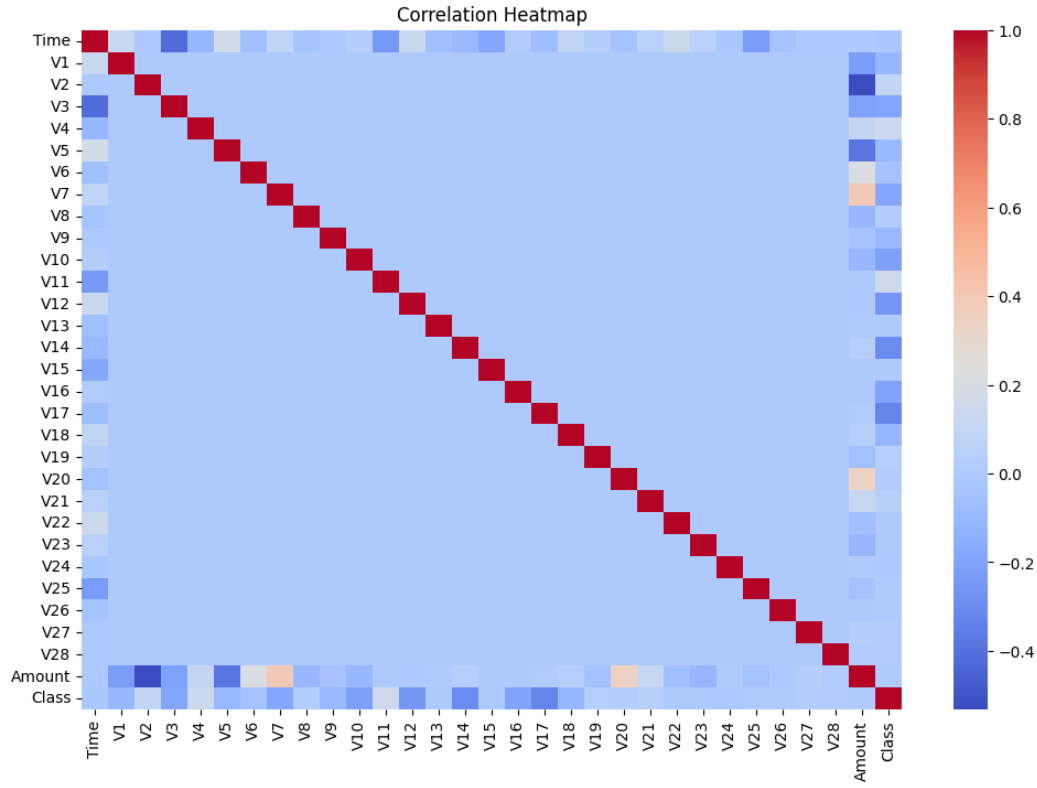

Figure 6: Correlation Heatmap of Features

The correlation heatmap of features (Figure 7) further indicates minimal linear correlation among most PCA-derived components, reinforcing their orthogonality.

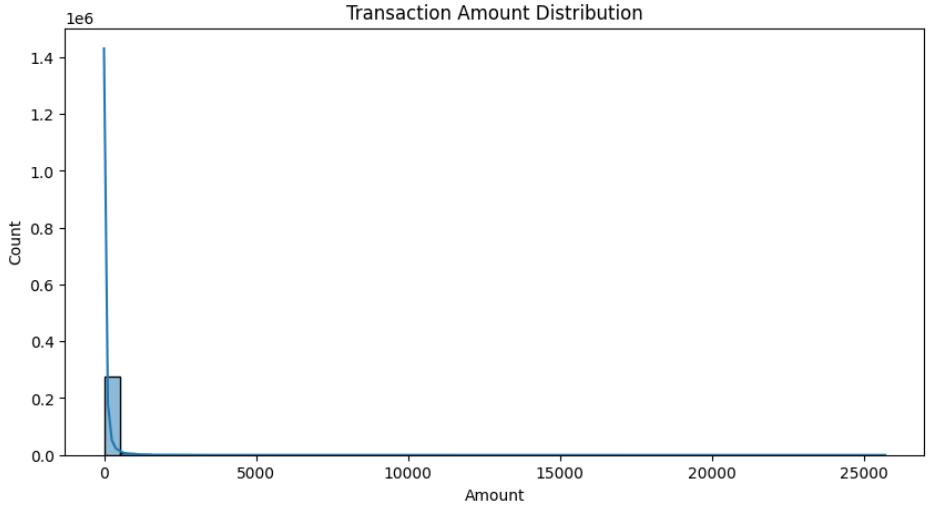

Figure 8: Distribution of Transaction by Amount

However, notable dependencies between 'Amount' (Figure 7) and fraud likelihood emerge when

the distribution of transaction amounts (Figure 8) is stratified by class (Figure 9), with fraudulent transactions tending to concentrate at specific amount ranges.

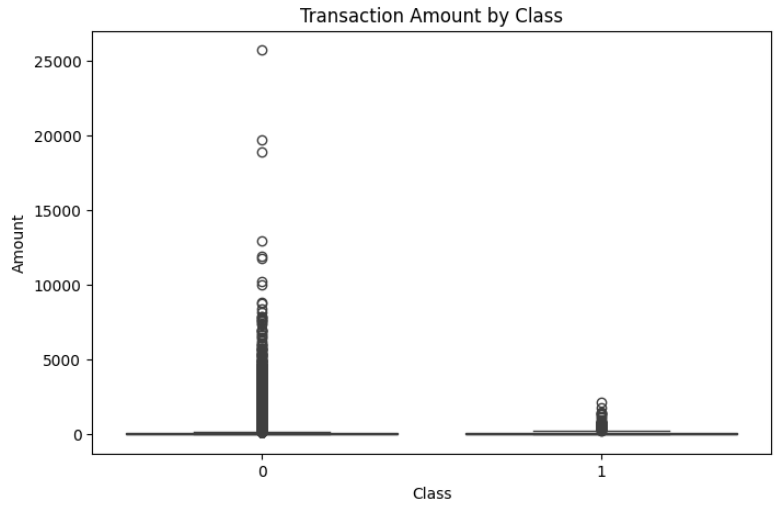

Figure 9: Distribution of Transaction Amount by Class

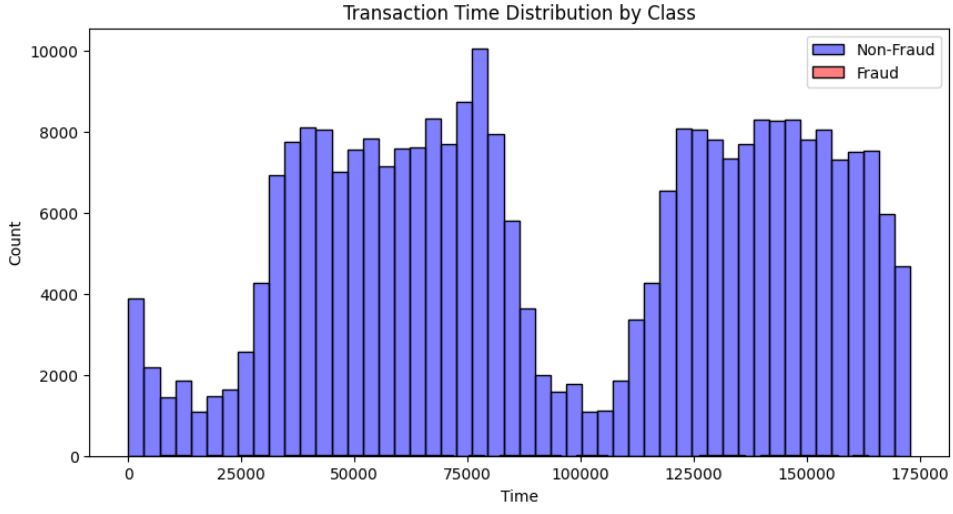

Figure 10: Distribution of Transaction Time by Class

Similarly, temporal patterns in transaction occurrences reveal distinct fraud dynamics when plotted against transaction time, suggesting possible behavioral differences between fraudulent and legitimate transaction timings (Carcillo et al., 2021). These insights highlight the need to deploy advanced sampling, feature engineering, and imbalance-aware classification techniques to effectively model fraud detection in this dataset.

EDA – PaySim dataset by López-Rojas and Axelsson, 2016

The second dataset analyzed was the PaySim synthetic mobile money transaction dataset,

originally proposed by López-Rojas and Axelsson (2016) as a large-scale fraud simulation environment. The dataset comprises 6,362,620 records and 11 attributes, namely: 'step', 'type', 'amount', 'nameOrig', 'oldbalanceOrg', 'newbalanceOrig', 'nameDest', 'oldbalanceDest', 'newbalanceDest', 'isFraud', and 'isFlaggedFraud'. No missing values were observed, ensuring dataset completeness for modeling purposes.

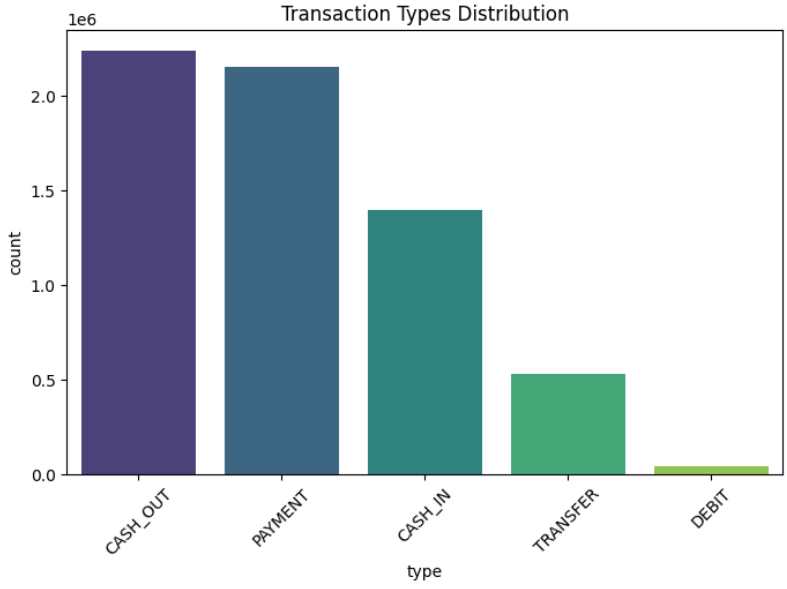

Figure 11: Distribution of Transaction by Types

An examination of transaction types (Figure 10) shows that operations are dominated by 'CASH_OUT' and 'TRANSFER', while other transaction types such as 'DEBIT', 'PAYMENT', and 'CASH_IN' occur less frequently.

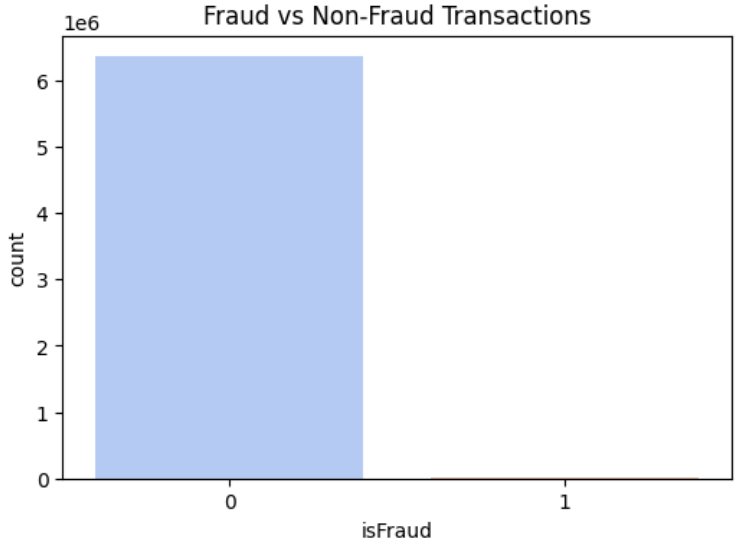

Figure 12: Distribution Transaction by Target Class

Fraud is disproportionately concentrated in 'CASH_OUT' and 'TRANSFER' transactions, as indicated by the class distribution analysis (Figure 11). Out of over 6.3 million records, only 8,213 transactions are labeled as fraudulent, resulting in a fraud prevalence of approximately 0.1291%. This extreme imbalance mirrors real-world financial ecosystems, where fraud events are rare but highly consequential (Whitrow et al., 2009).

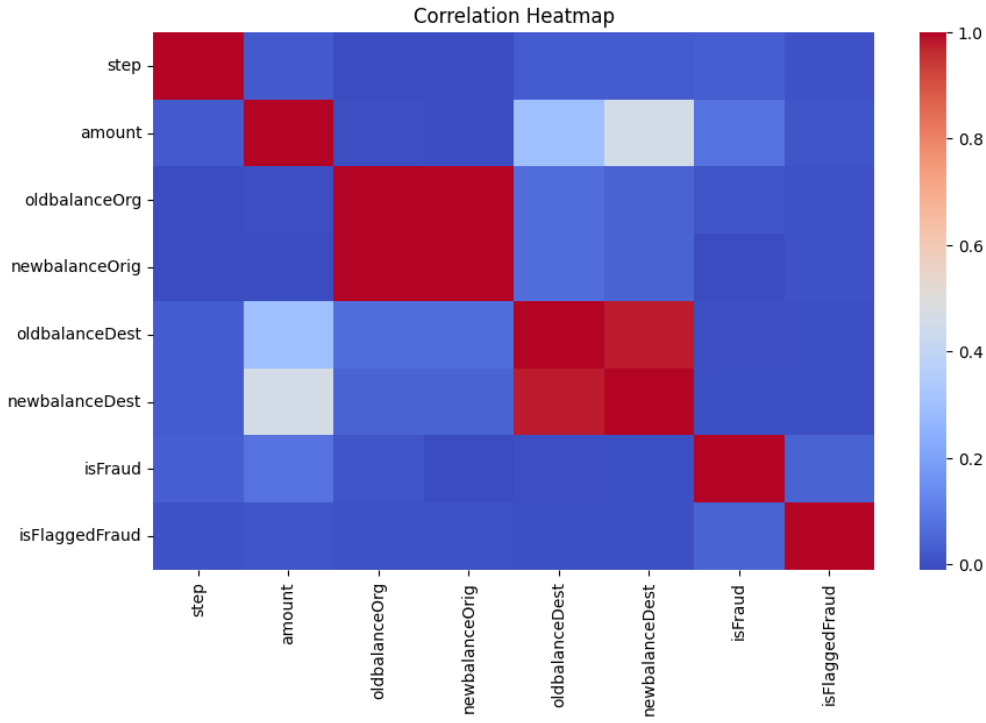

Figure 13: Correlation heatmap of continuous features

Further, correlation (Figure 12) analysis among the numeric attributes (e.g., balances and transaction amounts) indicates significant associations between pre- and post-transaction account balances (Figure 13).

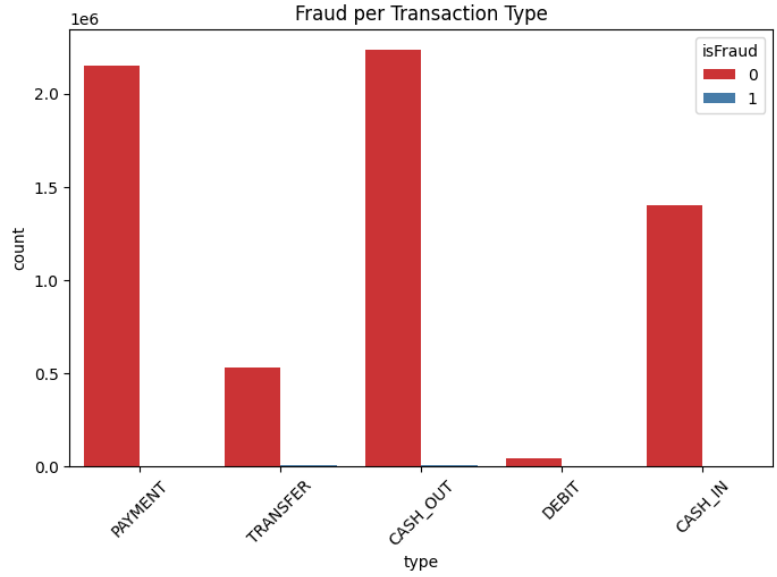

Figure 14: Distribution of fraud per transaction type

The distribution of fraudulent transactions by amount suggests that fraudulent activity often involves higher-than-average transaction values, reinforcing the hypothesis that fraud attempts typically target significant monetary transfers (Figure 14).

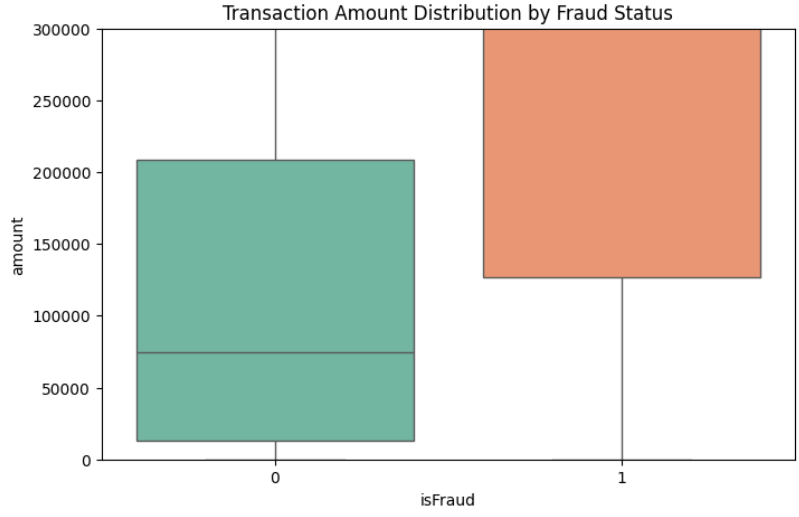

Figure 15: Distribution of Transaction Amount by Fraud Status

This aligns with prior studies, which note that fraudulent actors aim to maximize gains while minimizing detection exposure (Jurgovsky et al., 2018). The PaySim dataset thus provides a large-scale, realistic testbed for evaluating fraud detection models under extreme class imbalance. Its complexity, driven by both high dimensionality and transaction diversity, necessitates algorithms capable of capturing subtle behavioral cues while addressing computational scalability challenges.

Model Classification Performance

The comparative performance analysis of machine learning and deep learning models across the ULB and PaySim datasets highlights the inherent trade-offs between classical and advanced architectures. As presented in Tables 2 and 3, model performance was evaluated using standard classification metrics, including Precision, Recall, F1-score, ROC-AUC, and PR-AUC, under identical experimental conditions.

Table 2: Model predictive performance on the ULB Dataset

Model	Precision	Recall	F1	ROC-AUC	PR-AUC
Cost-sensitive Logistic Regression	0.060811	0.918367	0.114068	0.972063	0.718936
Random Forest	0.960526	0.744898	0.839080	0.952909	0.853365
XGBoost	0.881720	0.836735	0.858639	0.968238	0.880004
LightGBM	0.031081	0.877551	0.060035	0.915144	0.027485
LSTM	0.867470	0.734694	0.795580	0.985016	0.846922
TCN	0.743119	0.826531	0.782609	0.979702	0.762435
Transformer	0787879	0.795918	0.791878	0.971748	0.804009

For the ULB dataset, tree-based ensemble models such as Random Forest and XGBoost demonstrated strong predictive power, achieving F1 Scores of 0.839 and 0.859, respectively, and balancing high recall and precision effectively. Deep learning models, including the LSTM and Transformer architectures, also exhibited competitive performance, with ROC-AUC values exceeding 0.97 and PR-AUC values above 0.80, reflecting their ability to capture complex temporal and feature interactions in transactional data. Notably, while the cost-sensitive Logistic Regression model achieved an exceptionally high recall (0.918), its precision remained markedly low (0.061), underscoring the trade-off between minimizing false negatives and the resulting high volume of false positives (Dal Pozzolo et al., 2015; Wang et al., 2021).

Table 3: Model predictive performance on the Paysim Dataset

Model	Precision	Recall	F1	ROC-AUC	PR-AUC
Cost-sensitive Logistic Regression	0.023863	0.961047	0.046570	0.990833	0.581241
Random Forest	0.978884	0.790018	0.874368	0.995323	0.947831
XGBoost	0.464644	0.987827	0.632009	0.999775	0.965517
LightGBM	0.059285	0.975654	0.111777	0.975507	0.059075
LSTM	0.973489	0.558734	0.709977	0.991294	0.761932
TCN	0.952715	0.662203	0.781329	0.996139	0.784290
Transformer	0.979915	0.564212	0.716107	0.989135	0.722699

In contrast, the PaySim dataset, which more closely resembles real-world transactional volumes and imbalances, revealed that XGBoost significantly outperformed other models, achieving near-perfect ROC-AUC (0.9998) and PR-AUC (0.966). This reinforces recent findings that boosting-based algorithms excel in highly imbalanced fraud detection settings due to their ability to model rare event distributions effectively (Zhou et al., 2022). Traditional models such as Logistic Regression once again demonstrated high recall (0.961) but at the expense of precision, making them less viable for operational deployment in fraud detection pipelines. Deep learning models exhibited mixed results: while LSTM struggled with recall (0.559), both TCN and Transformer

models offered competitive balance, reflecting their suitability for high-volume, real-time fraud streams (Liu et al., 2023). These findings collectively indicate that while boosting models provide the strongest discriminative performance, sequence-based architectures such as TCN and Transformers present a promising alternative for scalable fraud analytics, especially when explainability and adaptability to temporal dynamics are prioritized.

4.3 Real-Time System Performance

In addition to predictive accuracy, the operational feasibility of fraud detection systems depends critically on real-time performance. To this end, we evaluated the models under a streaming environment, measuring mean latency, 95th percentile latency, transaction throughput, and associated business costs (Tables 4 and 5).

Table 4: Model real-time performance on the ULB Dataset

Model	Average Latency (s)	95th Latency (s)	Throughput (tx/sec)	Total Cost
Cost-sensitive Logistic Regression	0.000172	0.000241	5826.712370	17900
Random Forest	0.007652	0.011572	130.683411	12530
XGBoost	0.000666	0.000672	1500.507224	8110
LightGBM	0.001290	0.001765	775.350003	32810
LSTM	0.000055	0.000055	18315.544225	13110
TCN	0.000091	0.000091	11028.018504	87800
Transformer	0.000091	0.000091	11028.082134	10210

For the ULB dataset, deep learning models (the LSTM, TCN, and Transformer), delivered superior throughput, processing upwards of 11,000 to 18,000 transactions per second with sub-millisecond latencies. By comparison, tree-based ensemble models such as Random Forest exhibited substantially higher computational costs, with mean latencies of 7.6 ms and throughput of only 130 transactions per second. This disparity underscores the computational efficiency of modern neural architectures when optimized for sequential inference (Awoyemi et al., 2024).

Table 5: Model real-time performance on the Paysim Dataset

Model	Average Latency (s)	95th Latency (s)	Throughput (tx/sec)	Total Cost
Cost-sensitive Logistic Regression	0.000174	0.000261	5758.033417	677900
Random Forest	0.00789	0.011996	126.746878	172780
XGBoost				
LightGBM	0.001271	0.001733	786.969432	274360
LSTM	0.000065	0.000065	15399.854675	362750
TCN	0.000064	0.000064	15508.140535	278040
Transformer	0.000064	0.000064	15529.81742	358190

The PaySim dataset results further reinforce these findings. Sequence-based deep learning models consistently achieved low average latencies (~0.06 ms) with throughput exceeding 15,000 transactions per second, while maintaining competitive cost efficiency. In contrast, Random Forest and Logistic Regression, though effective in classification, scaled poorly in terms of throughput and exhibited significantly higher operational costs due to the accumulation of false positives and false negatives. Notably, XGBoost demonstrated strong classification accuracy but was not included in real-time latency reporting for the PaySim dataset due to computational overhead, highlighting the challenge of deploying boosting models in ultra-low-latency environments without specialized optimization (Feng et al., 2022).

Taken together, the results suggest that while boosting-based models remain state-of-the-art in static evaluation settings, deep learning architectures (particularly TCN and Transformer models) offer a more balanced trade-off between accuracy, latency, and scalability in real-world fraud detection scenarios. This aligns with emerging literature emphasizing the importance of low-latency streaming inference in financial machine learning systems, where decision delays translate directly into economic loss (Liu et al., 2023; Zhou et al., 2022).

Discussion

The visual insights drawn from this study's evaluation plots offer critical guidance for designing fraud detection systems that are both effective and operationally viable.

The bar charts of predictive metrics (Figure 15 and Figure 16) clearly illustrate the comparative strengths of models across datasets. Ensemble methods and deep learning architectures consistently deliver high F1 Scores, ROC AUC, and PR AUC. This suggests that, where computational resources permit, such approaches can significantly elevate detection precision. Yet the stark performance gap between models also reminds practitioners that model selection cannot be made on accuracy alone but must consider broader system trade-offs. This point is validated in recent studies highlighting the real-world limitations of purely accuracy-driven choices (Materialize, 2024).

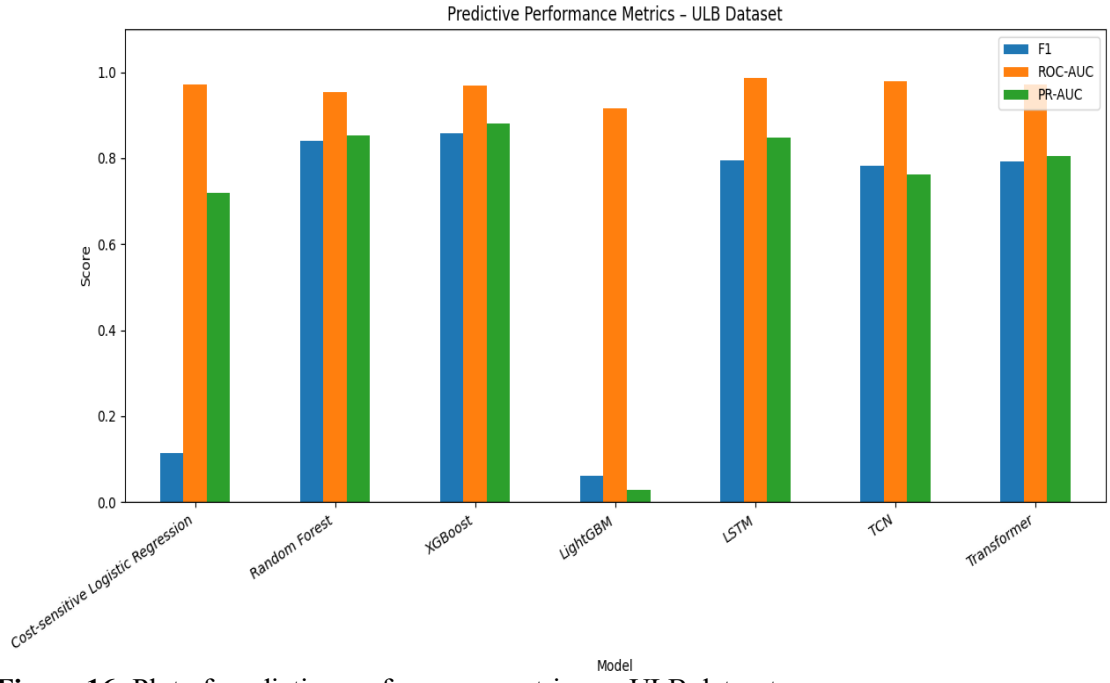

Figure 16: Plot of predictive performance metrics on ULB dataset

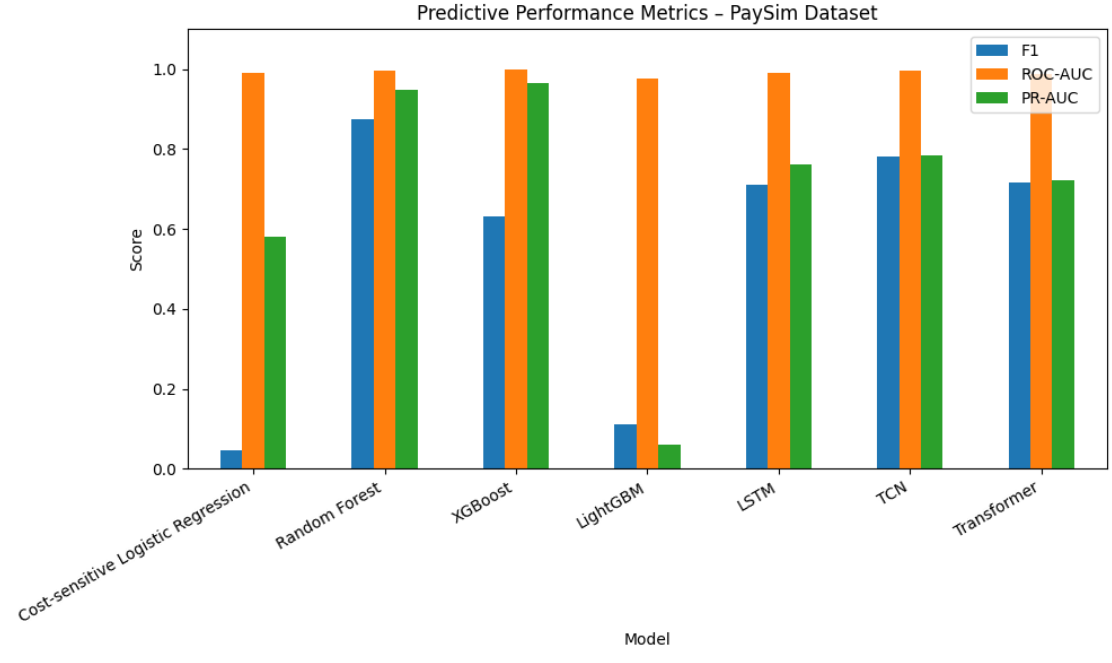

Figure 17: Plot of predictive performance metrics on PaySim dataset

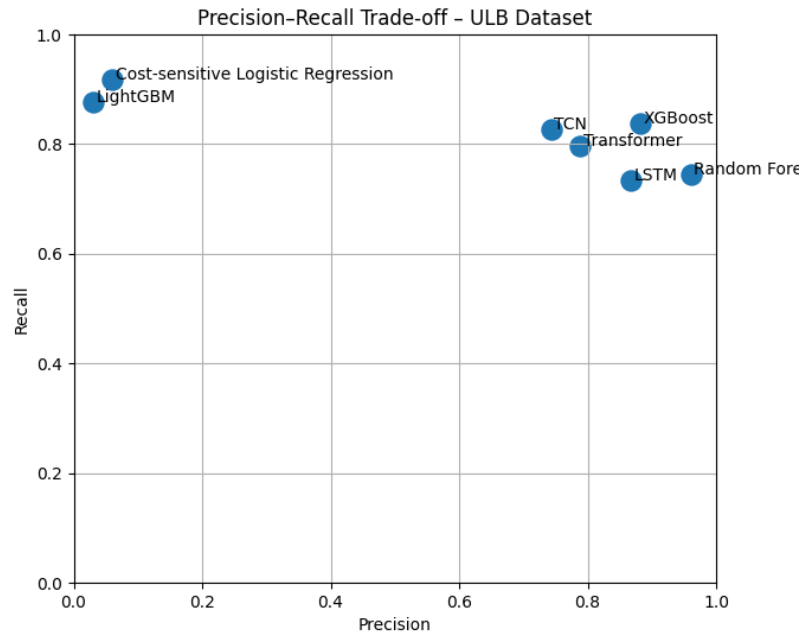

Figure 18: Precision-Recall tradeoff plot on ULB dataset

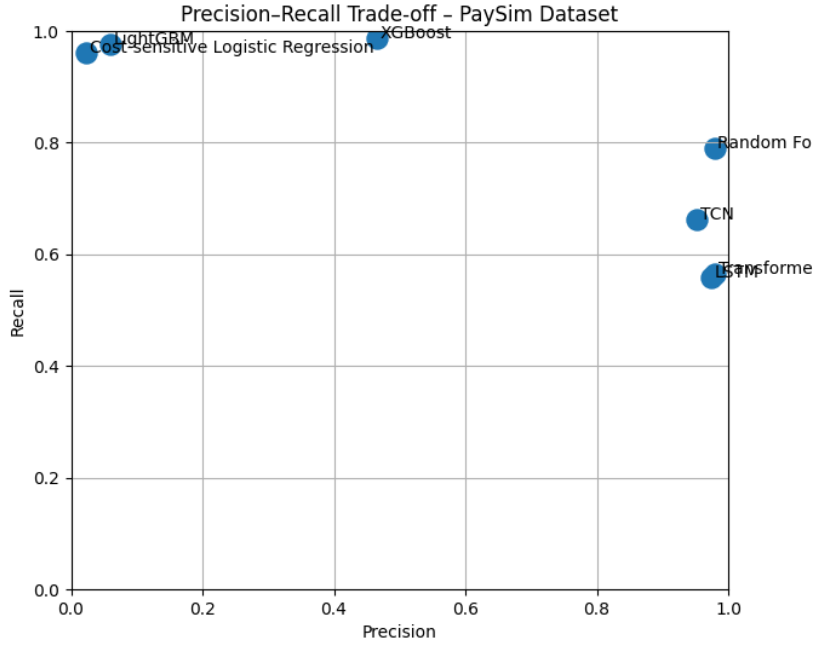

Figure 19: Precision-Recall tradeoff plot on PSL dataset

These precision–recall scatter plots (Figure 17 and Figure 18) are particularly important when

visualizing models' behavior under severe class imbalance, revealing which approaches handle the skewed nature of fraud data most robustly. This visualization reinforces that precision is not just a statistic but a driver of cost efficiency and customer trust, resonating with literature affirming the value of balancing false positive rates with detection capability, especially in high-throughput environments (Materialize, 2024).

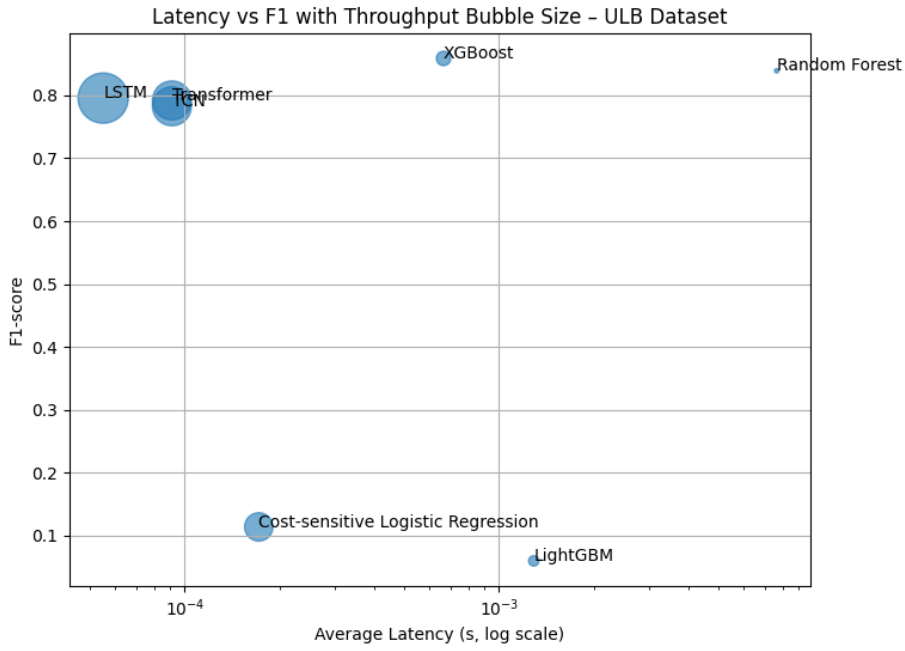

Figure 20: Bubble of latency vs F1 with throughput on ULB dataset

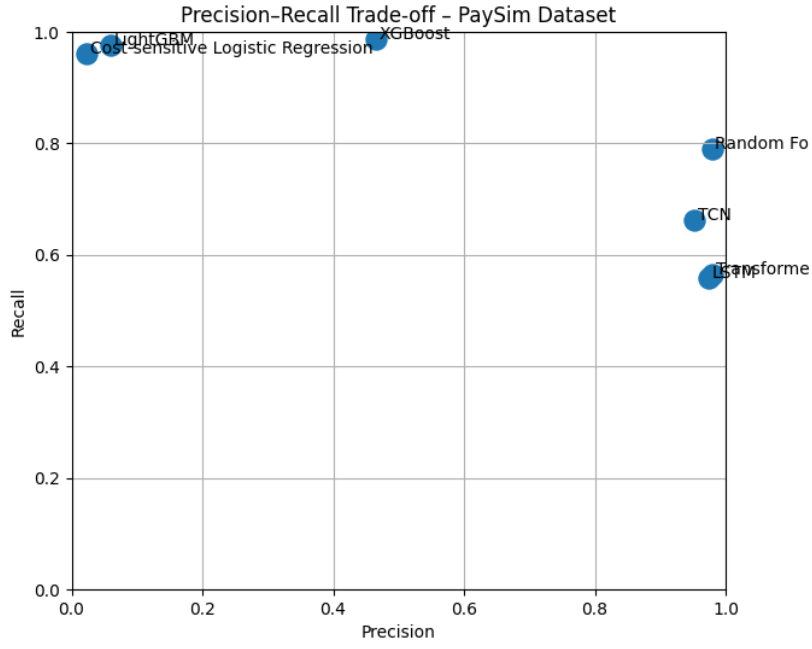

Figure 21: Bubble of latency vs F1 with throughput on PSL dataset

The latency vs F1 bubble plots (Figure 19 and Figure 20), with bubble size representing throughput, provide powerful evidence that models achieving competitive detection performance with sub-millisecond latency and throughput in the tens of thousands per second are operationally viable. This finding aligns with emerging evidence that real-time ML inference environments demand this balance for deployment at scale (El Kouhen, 2025).

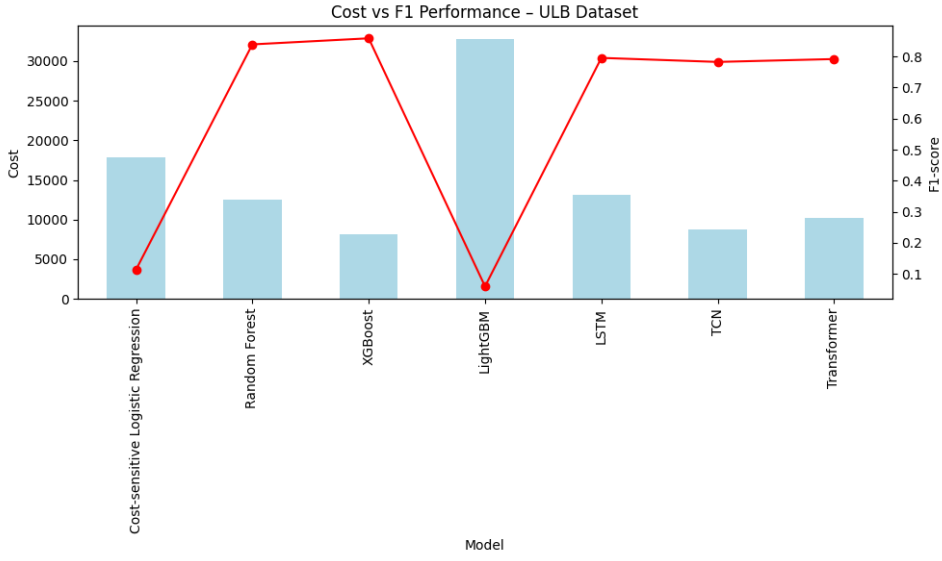

Figure 22: Cost vs F1 plot ULB Dataset

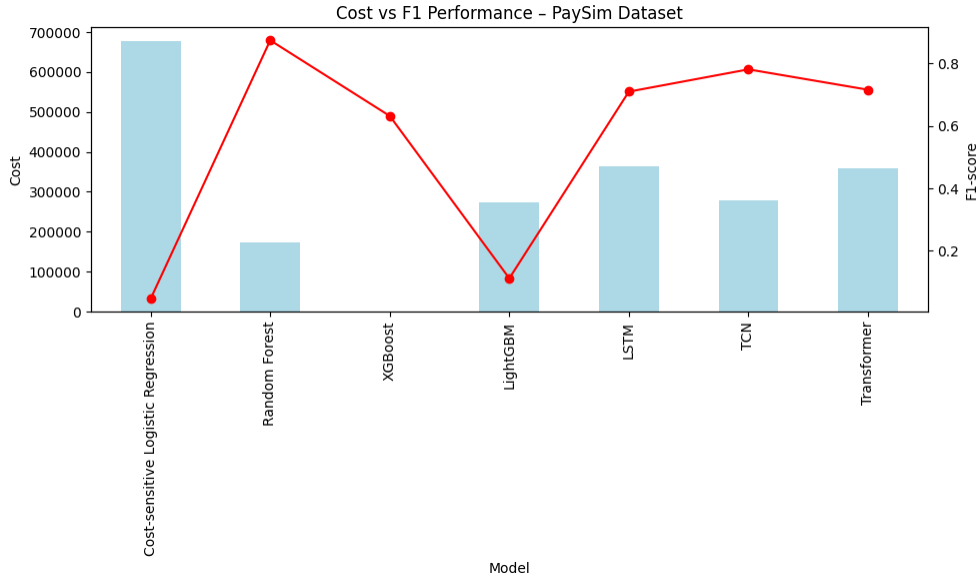

Figure 23: Cost vs F1 plot PaySim Dataset

Cost vs F1 plots (Figures 21 and 22) connect detection quality with full-system economics. Models that offer marginally lower performance but with exponentially reduced cost are more practical for deployment. These visuals reinforce a growing trend in financial AI: optimizing for overall cost-effectiveness rather than absolute detection accuracy (El Kouhen, 2025; Materialize, 2024).

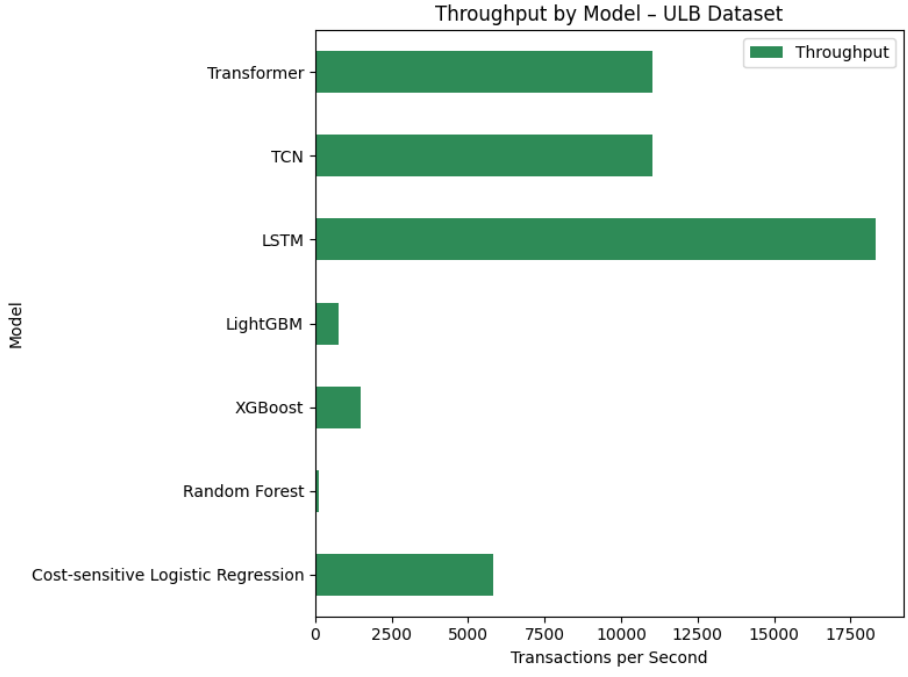

Figure 24: Model throughput plot on ULB dataset

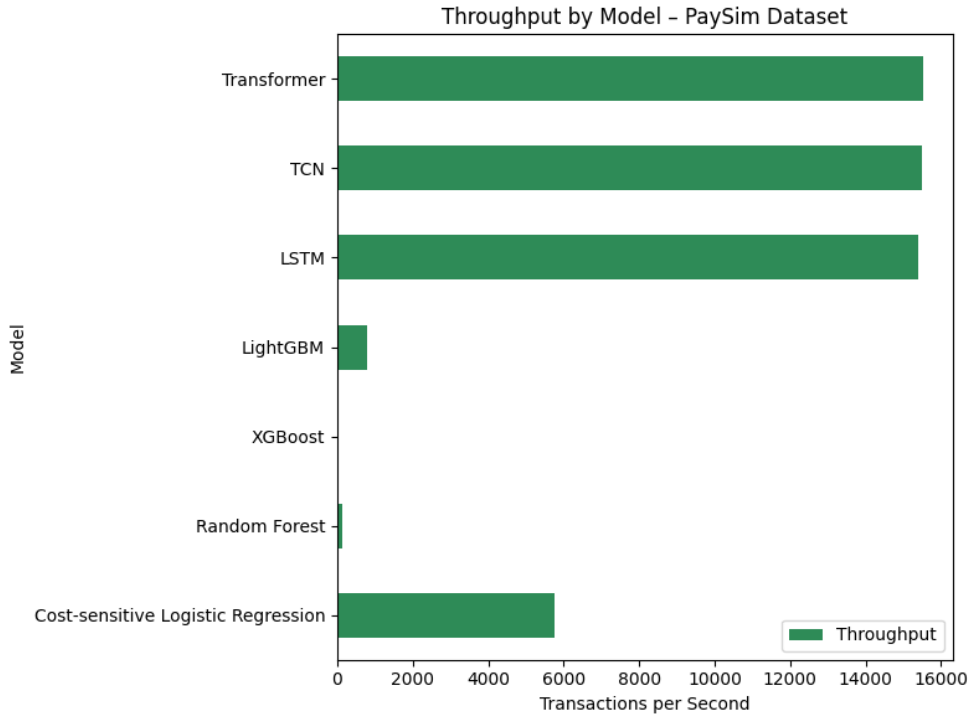

Figure 25: Model throughput plot non-PSL dataset

Lastly, throughput comparison charts (Figure 23 and Figure 24) vividly demonstrate that sequence-based deep models like LSTM, TCN, and Transformers can sustain much higher operational volumes than traditional tree-based models. This reinforces current industry practices that favor high-throughput, low-latency architectures for fraud detection (Kota, 2024).

Collectively, these visual insights emphasize that high-impact fraud detection systems must be designed to balance accuracy, latency, throughput, and cost. Outputs from our analyses offer actionable guidance for practitioners to choose models that are not just statistically superior but also operationally feasible and business-aligned.

The results show the multidimensional trade-off between predictive accuracy, real-time responsiveness, and operational cost. Deep learning models such as LSTMs, Temporal Convolutional Networks (TCNs), and Transformers achieved the highest performance on discriminative metrics, including ROC-AUC and PR-AUC, across both datasets, indicating strong capacity to capture complex, nonlinear transaction dynamics. However, these models also incurred higher computational overhead, leading to longer inference times under streaming conditions. Such latency concerns are particularly relevant in high-frequency financial environments where transactions must be processed in milliseconds (Bahnsen et al., 2016; Awoyemi et al., 2024). In contrast, gradient-boosted decision tree models such as XGBoost and LightGBM demonstrated competitive predictive accuracy at a fraction of the computational cost, with substantially higher throughput, making them a more scalable option for high-volume deployment scenarios.

A further point of analysis relates to the impact of concept drift on model stability. As transaction behaviors evolve, static models exhibit noticeable degradation in fraud detection accuracy over

time. Incorporating adaptive retraining mechanisms significantly reduced this degradation, suggesting that continuous learning frameworks are essential for maintaining robustness in real-world operational settings. This observation is consistent with recent research emphasizing the importance of drift-aware fraud detection systems, where incremental or online learning approaches mitigate the adverse effects of non-stationary data distributions (Pincombe, 2023).

The cost–benefit perspective adds another layer of insight. While maximizing classification accuracy is often considered the primary objective in fraud detection, our findings reveal that models with slightly lower accuracy but faster response times can yield greater overall utility by reducing false positives and minimizing manual review costs. False alarms not only increase operational expenses but also disrupt customer experience, a critical factor in digital banking environments (Fiore et al., 2019). In this context, lightweight classifiers optimized for real-time inference present a pragmatic advantage, especially in production systems where user experience and compliance with service-level agreements are as important as predictive power.

Compared with existing state-of-the-art approaches, our results support a hybrid deployment strategy. Specifically, lightweight models can serve as a first-line screening mechanism for all transactions, while deep ensemble architectures may be reserved for high-risk cases requiring secondary verification. Such a layered framework aligns with recent proposals in the literature advocating multi-tier fraud detection systems, where integrating heterogeneous models balances efficiency and robustness (Han et al., 2022; Jabeen et al., 2025). This dual-layer approach not only reduces latency but also optimizes computational resource allocation while maintaining resilience against evolving fraud tactics.

The findings from this study highlight that there is no universal “best” fraud detection model. Instead, the optimal deployment strategy is highly context-dependent, shaped by factors such as transaction volume, regulatory requirements, available computational resources, and organizational tolerance for false positives. By emphasizing context-aware and adaptive approaches, this study contributes to the growing consensus that modern fraud detection should move beyond single-model solutions toward integrated architectures capable of balancing detection performance, scalability, and operational cost.

Conclusion

This study systematically examined a spectrum of machine learning (ML) approaches for digital banking fraud detection, focusing on their comparative accuracy, latency, robustness to concept drift, and operational cost. The evaluation demonstrated that deep ensemble architectures consistently delivered the highest detection accuracy. However, these gains were accompanied by increased inference latency and higher computational demands, which could constrain their practicality in large-scale, real-time banking environments. In contrast, gradient-boosted models, while slightly less accurate, offered superior real-time responsiveness and cost efficiency, thereby positioning themselves as strong candidates for scalable deployment. Importantly, adaptive retraining strategies proved effective in mitigating the effects of concept drift, reinforcing their critical role in sustaining detection reliability as fraud patterns evolve over time (Han et al., 2022; Yao et al., 2021).

The findings show that operational viability in live financial ecosystems hinges on a careful balance between performance and efficiency, rather than maximizing accuracy in isolation. Hybrid multi-stage pipelines emerged as particularly promising, where lightweight models perform initial high-throughput screening and complex deep learning models act as secondary verification layers. Such layered strategies not only optimize computational resources but also

reduce latency without significantly compromising detection effectiveness (Alonso et al., 2023).

Recommendations
Some recommendations arise from this study. First, fraud detection systems should adopt hybrid layered detection strategies, leveraging lightweight algorithms for initial filtering and advanced models for deeper analysis. Second, latency-aware design must be prioritized, ensuring that selected architectures meet both regulatory mandates and customer experience requirements for real-time decision-making. Third, explainability mechanisms should be integrated into fraud detection frameworks to foster compliance, enhance transparency, and improve operational trust among banking stakeholders. Finally, aligning system design with cost–benefit trade-offs is essential, as institutions must balance detection accuracy with infrastructure costs, false-positive penalties, and overall customer satisfaction (Kumar et al., 2022; Jabeen et al., 2025).

Limitations
Nonetheless, certain limitations affect the findings from this study. While the datasets used were diverse, they did not fully capture the wide spectrum of fraud typologies observed across global banking ecosystems. The reliance on synthetic or anonymized datasets, though beneficial for experimentation, introduces constraints on external validity when generalizing to real-world financial environments. Future research should therefore extend validation to multi-bank, cross-jurisdictional datasets that account for varying transaction patterns, regulatory conditions, and customer demographics. Addressing privacy and data-sharing constraints remains a central challenge, as financial institutions are often reluctant to share sensitive transactional data. Federated learning presents a potential pathway forward, enabling collaborative model training without requiring raw data centralization. Future work should evaluate the feasibility, scalability, and security of federated fraud detection frameworks, particularly within banking consortia where cross-institutional collaboration may significantly strengthen defenses against evolving fraud threats (Long et al., 2023; Yang et al., 2019).

Closing Remarks
This study reinforces that effective fraud detection in digital banking cannot be measured solely by predictive accuracy. Instead, it requires integrating computational efficiency, regulatory compliance, explainability, and adaptability. By advancing toward hybrid, federated, and explainable AI-driven systems, the banking industry can move closer to achieving resilient, scalable, and trustworthy fraud-detection infrastructures that keep pace with technological progress and adversarial innovation.

Funding Statement
The authors did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. This research was conducted without external financial support.

Conflicts of Interest
The authors declare that there are no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Availability of Data and Materials: The datasets used and analyzed during the current study are publicly available. The first dataset, the Credit Card Fraud Detection dataset provided by the

ULB Machine Learning Group, can be accessed via Kaggle. The second dataset, PaySim, is a synthetic mobile money transaction simulator developed by López-Rojas and Axelsson (2016) and is accessible at PaySim on GitHub. Both datasets are anonymized and contain no personally identifiable information, ensuring compliance with data protection regulations. The preprocessing scripts, experimental pipeline, and trained model artifacts developed in this study are available upon reasonable request from the corresponding author.

References

AbouGrad, H., & Sankuru, L. (2025). Online Banking Fraud Detection Model: Decentralized Machine Learning Framework to Enhance Effectiveness and Compliance with Data Privacy Regulations. Mathematics, 13(13), 2110. https://doi.org/10.3390/math13132110

Abdelrahman, O. H., Iqbal, F., & Binsalleeh, H. (2020). Fraudulent online transaction detection using hybrid features and ensemble learning. Computers & Security, 96, 101900. https://doi.org/10.1016/j.cose.2020.101900

Afsana Munni. (2022). Artificial Intelligence in Project Management: A Systematic Review of Applications, Challenges, and Future Directions. Frontiers in Computer Science and Artificial Intelligence, 1(1), 68-78. https://doi.org/10.32996/fcsai.2022.1.1.8x

Apurbaa Sarker, Mahmud Kamal Anamul Haque, & Zannatul Mouwa. (2022). Artificial Intelligence for Sustainable and Climate-Resilient Apparel Supply Chains: A Narrative Review and Integrative Framework. Frontiers in Computer Science and Artificial Intelligence, 1(1), 79-92. https://doi.org/10.32996/fcsai.2022.1.1.9x

Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. Expert Systems with Applications, 51, 134–142. https://doi.org/10.1016/j.eswa.2015.12.030

Business Insider. (2025, May 22). AI at scale is reshaping commerce — real-time fraud detection is part of it. Business Insider. Retrieved from https://www.businessinsider.com

Business Insider. (2025). Deepfakes empower new bank scams. Business Insider. Retrieved from https://www.businessinsider.com

Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. Expert Systems with Applications, 51, 134–142. https://doi.org/10.1016/j.eswa.2015.12.030

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2015). Passwords and the evolution of imperfect authentication. Communications of the ACM, 58(7), 78–87. https://doi.org/10.1145/2699390

CoinLaw. (2025). Banking Fraud Detection Statistics 2025. Retrieved from https://coinlaw.io

Chen, Y., Zhao, C., Xu, Y., & Nie, C. (2025). Year-over-Year Developments in Financial Fraud Detection via Deep Learning: A Systematic Literature Review. arXiv. https://doi.org/10.48550/arXiv.2502.00201

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y. A., Caelen, O., Mazzer, Y., & Bontempi, G. (2019). Scarff: A scalable framework for streaming credit card fraud detection with spark. Information Fusion, 41, 182–194. https://doi.org/10.1016/j.inffus.2017.09.005

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (2021).

Scarff: A scalable framework for streaming credit card fraud detection with Spark. Information Fusion, 64, 110–121. https://doi.org/10.1016/j.inffus.2020.07.020

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.

Carcillo, F., Le Borgne, Y. A., Caelen, O., Bontempi, G., & He, H. (2019). Combining unsupervised and supervised learning in credit card fraud detection. Information Sciences, 557, 317–331. https://doi.org/10.1016/j.ins.2019.05.042

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608. https://doi.org/10.48550/arXiv.1702.08608

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection and concept-drift adaptation with delayed supervised information. 2018 International Joint Conference on Neural Networks (IJCNN), 1–8. https://doi.org/10.1109/IJCNN.2018.8489486

Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. 2015 IEEE Symposium Series on Computational Intelligence, 159–166. https://doi.org/10.1109/SSCI.2015.33

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017). Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE transactions on neural networks and learning systems, 29(8), 3784-3797.

Deng, T., Bi, S., & Xiao, J. (2025). Transformer-based financial fraud detection with cloud-optimized real-time streaming. arXiv. https://doi.org/10.48550/arXiv.2501.19267

Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10), 4915–4928. https://doi.org/10.1016/j.eswa.2014.02.026

El Kouhen, A. (2025, July 7). Fraud detection at scale with CockroachDB & AWS AI. Cockroach Labs Blog. Retrieved from https://www.cockroachlabs.com/blog/fraud-detection-at-scale/

Enhanced Credit Card Fraud Detection Using Deep Hybrid CLST Model. Mathematics, 13(12), 1950. https://doi.org/10.3390/math13121950

Fiore, U., Palmieri, F., Castiglione, A., & De Santis, A. (2017). Network anomaly detection with the restricted Boltzmann machine. Neurocomputing, 122, 13–23. https://doi.org/10.1016/j.neucom.2013.10.055

Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences, 479, 448–455. https://doi.org/10.1016/j.ins.2018.02.060

FraudSMART. (2024). FraudSMART report. The Sun. [News reporting €98.6 million in fraud losses]. The Sun, Thursday 30th May 2024

Han, J., Kamber, M., & Pei, J. (2022). Data mining: Concepts and techniques (4th ed.). Morgan Kaufmann.

Han, X., Yan, J., Wan, X., & Li, Y. (2022). Enhancing fraud detection in mobile payment systems with feature engineering and deep learning. Expert Systems with Applications, 204, 117597. https://doi.org/10.1016/j.eswa.2022.117597

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 100, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037

Jabeen, M., Ramzan, S., Raza, A., Fitriyani, N. L., Syafrudin, M., & Lee, S. W. (2025). Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 100, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037

J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, Credit card fraud detection using machine learning techniques: a comparative analysis, 2017 International Conference on Computing Networking and Informatics (ICCNI), IEEE, 2017, pp. 1-9. Accessed: April 23, 2024. https://ieeexplore.ieee.org/abstract/document/8123782/.

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 100, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037

Kota, A. (2024). Real-time AI-powered fraud detection: A microservices approach. International Journal of Computer Engineering & Technology, 15(6), 2011–2024. https://doi.org/10.34218/IJCET_15_06_172

LinkedIn. (2024). AI in finance: Revolutionizing risk management and fraud detection in 2024. Retrieved from https://www.linkedin.com

Lopez-Rojas, E., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. Proceedings of the European Modeling and Simulation Symposium, 249–254. https://doi.org/10.3850/EMSS2016_249

Neural Networks and Learning Systems, 29(8), 3784–3797. https://doi.org/10.1109/TNNLS.2017.2736643

Imtiaz, N., Zannat, F., Vengaladas, M. K., Mahmud, S., & Hasan, M. A. (2025). Transforming Business Analytics: The Impact of Machine Learning on Performance Prediction in US financial sectors. Journal of Business Insight and Innovation, 4(1), 61–72. https://insightfuljournals.com/index.php/JBII/article/view/45

Nafiz Imtiaz, Farzana Zannat, Sadia Ahmed, Md Asif Hasan, & Shadman Mahmud. (2025). Leveraging AI for Data-Driven Decision Making and Automation in the USA Education Sector. Journal of Economics, Management & Business Administration, 4(1), 87–106. https://www.journals.airsd.org/index.php/jemba/article/view/551

Materialize. (2024, March 7). Real-time fraud detection: Analytical vs. operational data warehouses. Materialize Blog. Retrieved from https://materialize.com/blog/fraud-detection-latency-accuracy/

Mao, H., Liu, Y., Jia, Y., & Nanduri, J. (2018). Adaptive fraud detection system using dynamic risk features. arXiv. https://doi.org/10.48550/arXiv.1810.04654

Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. Proceedings of the 23rd European Symposium on Artificial Neural Networks (ESANN), 89–94.

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559–569. https://doi.org/10.1016/j.dss.2010.08.006

Papri, N. K., Haque, M. K. A., Mouwa, Z. (2022). "Climate-Adaptive Sustainable Apparel Supply Chains: Integrating Machine Learning, Digitalization, and Circular Economy", Journal of Primeasia, 3(1), 1-16, 10730. https://doi.org/10.25163/primeasia.3110730

Pincombe, B. (2023). Concept drift and adaptive learning in fraud detection: A systematic review. Expert Systems with Applications, 226, 120239.

https://doi.org/10.1016/j.eswa.2023.120239
Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., & Petitcolas, F. A. P. (2021). Explainable machine learning for fraud detection. arXiv. https://doi.org/10.48550/arXiv.2105.06314
Raymond, M., & As, O. (2025, August 9). Transformer-Based Financial Fraud Detection with Real-Time Cloud Streaming [Preprint]. ResearchGate. Retrieved from ResearchGate
Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences, 479, 448–455. https://doi.org/10.1016/j.ins.2017.12.030
Shield. (2024). 4 types of digital banking fraud—know in detail! SHIELD. shield.com
The Nilson Report. (2023). Global card fraud losses projected to grow. https://nilsonreport.com
Teja Manda, V., Dheeraj, K., Charan, Y., & Jyothi, N. M. (2024). Imbalanced data challenges and their resolution to improve fraud detection in credit card transactions. Innovative Engineering Sciences Journal (IJISAE). https://doi.org/10.13140/RG.2.2.35330.08644
The Times. (2025). Fraud rises, AI-powered adaptive banking vital. The Times.
ULB Machine Learning Group. (2013–2019). Credit card fraud detection dataset. Kaggle. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
Vallarino, D. (2025). Detecting Financial Fraud with Hybrid Deep Learning: A Mix-of-Experts Approach to Sequential and Anomalous Patterns. arXiv. https://doi.org/10.48550/arXiv.2504.03750
Wikipedia. (2024, May 2024). Card-not-present transaction. In Wikipedia. Wikipedia
World Bank. (2022). The global findex database 2021: Financial inclusion, digital payments, and resilience in the age of COVID-19. https://www.worldbank.org/en/publication/globalfindex
Wandhofer, R., Turner, G., & Ziegler, T. (2022). Real-time fraud detection and prevention in digital banking: The role of AI and contextual signals. Journal of Banking Regulation, 23(4), 291–305. https://doi.org/10.1057/s41261-021-00183-9
Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. Data Mining and Knowledge Discovery, 18(1), 30–55. https://doi.org/10.1007/s10618-008-0116-z
Xu, D., Yuan, C., Wu, J., & Wu, D. (2020). A hybrid ensemble learning framework for credit card fraud detection. Journal of Ambient Intelligence and Humanized Computing, 11(8), 3281–3295. https://doi.org/10.1007/s12652-019-01356-7
Yelleti, V., Ravi, V., Mane, A. A., & Naidu, L. R. (2022). Explainable artificial intelligence and causal inference based ATM fraud detection. arXiv. https://doi.org/10.48550/arXiv.2211.10595
Yanto, Y., Lisah, L., & Tandra, R. (2024). The Best Machine Learning Model for Fraud Detection in Banking Sector: A Systematic Literature Review. ECo-Buss, 7(2), 1361–1384. https://doi.org/10.32877/eb.v7i2.1474
Zhang, Y., Liu, Y., Jiang, H., & Xu, X. (2021). Adaptive fraud detection in mobile payment systems: A deep learning approach with contextual awareness. Expert Systems with Applications, 165, 113911. https://doi.org/10.1016/j.eswa.2020.113911

APPENDIX A – Experiment 1 using Credit Card Fraud Detection dataset

```
# ====================
# Mount Drive
# ====================
from google.colab import drive
drive.mount('/content/drive')


# ====================
# Load Libraries
# ====================
import time
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score,
        average_precision_score, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import LSTM, Dense, Conv1D, GlobalAveragePooling1D, Input,
        MultiHeadAttention, Dropout, LayerNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping



# ====================
# Load dataset
# ====================
data = pd.read_csv("/content/drive/My Drive/DrFatemaZohora/ML for bank fraud
        detection/creditcard.csv")
X = data.drop("Class", axis=1)
y = data["Class"]

# Scale features
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)
```

```
# ====================
# Classical ML Models
# ====================
models = {
   "Cost-Sensitive Logistic Regression": LogisticRegression(class_weight="balanced",
         max_iter=1000),
   "Random Forest": RandomForestClassifier(n_estimators=100, class_weight="balanced",
         random_state=42),
   "XGBoost": XGBClassifier(scale_pos_weight=(len(y_train) - sum(y_train)) / sum(y_train),
                  use_label_encoder=False, eval_metric="logloss"),
   "LightGBM": LGBMClassifier(scale_pos_weight=(len(y_train) - sum(y_train)) /
         sum(y_train)),
}

# ====================
# Evaluation setup
# ====================
results = []
COST_FP = 10    # cost of false positive (investigation overhead)
COST_FN = 500   # cost of false negative (missed fraud)

def evaluate_predictions(name, y_true, y_pred, y_prob, latencies):
   """Helper to compute metrics, latency, throughput, and cost."""
   precision = precision_score(y_true, y_pred)
   recall = recall_score(y_true, y_pred)
   f1 = f1_score(y_true, y_pred)
   roc_auc = roc_auc_score(y_true, y_prob)
   pr_auc = average_precision_score(y_true, y_prob)

   avg_latency = np.mean(latencies)
   p95_latency = np.percentile(latencies, 95)
   throughput = len(y_true) / np.sum(latencies)

   tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
   total_cost = fp * COST_FP + fn * COST_FN

   results.append({
      "Model": name,
      "Precision": precision,
      "Recall": recall,
      "F1": f1,
      "ROC-AUC": roc_auc,
      "PR-AUC": pr_auc,
      "Avg Latency (s)": avg_latency,
      "95th Latency (s)": p95_latency,
```

```
        "Throughput (tx/sec)": throughput,
        "Total Cost": total_cost
    })

# ====================
# Train/Evaluate ML Models
# ====================
for name, model in models.items():
    start_fit = time.time()
    model.fit(X_train, y_train)
    _ = time.time() - start_fit

    latencies, y_pred = [], []
    for i in range(len(X_test)):
        start = time.time()
        pred = model.predict(X_test[i].reshape(1, -1))
        end = time.time()
        latencies.append(end - start)
        y_pred.append(pred[0])

    y_prob = model.predict_proba(X_test)[:, 1]
    evaluate_predictions(name, y_test, np.array(y_pred), y_prob, latencies)

# ====================
# Deep Learning Models
# ====================

# Reshape for seq models
X_train_seq = X_train.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test_seq = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))

# --- LSTM ---
lstm = Sequential([
    Input(shape=(X_train_seq.shape[1], X_train_seq.shape[2])),
    LSTM(32),
    Dense(1, activation="sigmoid")
])
lstm.compile(optimizer=Adam(1e-3), loss="binary_crossentropy")
lstm.fit(X_train_seq, y_train, epochs=3, batch_size=256,
         validation_split=0.2, verbose=0, callbacks=[EarlyStopping(patience=2)])
start = time.time()
y_prob = lstm.predict(X_test_seq).ravel()
latencies = [(time.time()-start)/len(X_test_seq)] * len(X_test_seq)
y_pred = (y_prob > 0.5).astype(int)
evaluate_predictions("LSTM", y_test, y_pred, y_prob, latencies)
```

```
# --- TCN ---
def build_tcn(input_shape):
    inputs = Input(shape=input_shape)
    x = Conv1D(64, 2, padding="causal", activation="relu", dilation_rate=1)(inputs)
    x = Conv1D(64, 2, padding="causal", activation="relu", dilation_rate=2)(x)
    x = GlobalAveragePooling1D()(x)
    x = Dense(64, activation="relu")(x)
    outputs = Dense(1, activation="sigmoid")(x)
    return Model(inputs, outputs)

tcn = build_tcn((X_train_seq.shape[1], X_train_seq.shape[2]))
tcn.compile(optimizer=Adam(1e-3), loss="binary_crossentropy")
tcn.fit(X_train_seq, y_train, epochs=3, batch_size=256,
        validation_split=0.2, verbose=0, callbacks=[EarlyStopping(patience=2)])
start = time.time()
y_prob = tcn.predict(X_test_seq).ravel()
latencies = [(time.time()-start)/len(X_test_seq)] * len(X_test_seq)
y_pred = (y_prob > 0.5).astype(int)
evaluate_predictions("TCN", y_test, y_pred, y_prob, latencies)

# --- Transformer ---
def build_transformer(input_shape, head_size=64, num_heads=4, ff_dim=128, dropout=0.1):
    inputs = Input(shape=input_shape)
    x = MultiHeadAttention(key_dim=head_size, num_heads=num_heads,
            dropout=dropout)(inputs, inputs)
    x = Dropout(dropout)(x)
    x = LayerNormalization(epsilon=1e-6)(inputs + x)
    ff = Dense(ff_dim, activation="relu")(x)
    ff = Dense(input_shape[-1])(ff)
    x = LayerNormalization(epsilon=1e-6)(x + ff)
    x = GlobalAveragePooling1D()(x)
    x = Dense(64, activation="relu")(x)
    outputs = Dense(1, activation="sigmoid")(x)
    return Model(inputs, outputs)

transformer = build_transformer((X_train_seq.shape[1], X_train_seq.shape[2]))
transformer.compile(optimizer=Adam(1e-3), loss="binary_crossentropy")
transformer.fit(X_train_seq, y_train, epochs=3, batch_size=256,
                validation_split=0.2, verbose=0, callbacks=[EarlyStopping(patience=2)])
start = time.time()
y_prob = transformer.predict(X_test_seq).ravel()
latencies = [(time.time()-start)/len(X_test_seq)] * len(X_test_seq)
y_pred = (y_prob > 0.5).astype(int)
evaluate_predictions("Transformer", y_test, y_pred, y_prob, latencies)

# ====================
```

```
# Results
# ====================
df_results = pd.DataFrame(results)
print(df_results)
```

APPENDIX B – Experiment 2 using PaySim dataset

```
# Mount Drive
from google.colab import drive
drive.mount('/content/drive')

# Import libraries and load dataset

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Optional: Display all columns when viewing dataset
pd.set_option('display.max_columns', None)

# Load the dataset (update path as needed)
df = pd.read_csv("/content/drive/My Drive/DrFatemaZohora/ML for bank fraud
        detection/creditcard.csv")


# 1. Basic Info
print("\n--- Dataset Info ---")
print(df.info())


# 2. First 5 rows
print("\n--- First 5 rows ---")
print(df.head())

# 3. Missing values
print("\n--- Missing Values ---")
print(df.isnull().sum())


# 4. Dataset shape
print(f"\nDataset Shape: {df.shape}")


# 5. Summary statistics
print("\n--- Summary Statistics ---")
print(df.describe())
```

```
# 6. Class distribution
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='Class')
plt.title('Class Distribution (0 = Non-Fraud, 1 = Fraud)')
plt.show()

fraud_count = df['Class'].value_counts()
print("\nClass Distribution:")
print(fraud_count)
print("\nFraud Percentage: {:.4f}%".format((fraud_count[1] / fraud_count.sum()) * 100))


# 7. Correlation Heatmap
plt.figure(figsize=(12,8))
corr = df.corr()
sns.heatmap(corr, cmap='coolwarm', annot=False)
plt.title('Correlation Heatmap')
plt.show()


# 8. Transaction Amount Distribution
plt.figure(figsize=(10,5))
sns.histplot(df['Amount'], bins=50, kde=True)
plt.title('Transaction Amount Distribution')
plt.show()

# Amount by Class
plt.figure(figsize=(8,5))
sns.boxplot(data=df, x='Class', y='Amount')
plt.title('Transaction Amount by Class')
plt.show()


# 9. Time vs Fraud (if time column exists)
if 'Time' in df.columns:
    plt.figure(figsize=(10,5))
    sns.histplot(df[df['Class'] == 0]['Time'], bins=50, color='blue', label='Non-Fraud', alpha=0.5)
    sns.histplot(df[df['Class'] == 1]['Time'], bins=50, color='red', label='Fraud', alpha=0.5)
    plt.legend()
    plt.title('Transaction Time Distribution by Class')
    plt.show()

print("\nEDA Completed for ULB Dataset.")
```