IAA Journal of Scientific Research 12(2):60-70, 2025. www.iaajournals.org
©IAAJOURNALS ISSN: 2636-7319
https://doi.org/10.59298/IAAJSR/2025/1226070.00 IAAJSR:122

Comparative Study of Real-Time and Batch Processing Approaches in Machine Learning-Based Fraud Detection for Financial Institutions

Egwu C. Victor, Akawuku I. Godspower and Adejumo O. Samuel

Department of Computer Science, Nnamdi Azikiwe University, Awka.
Email: egwudumebi@gmail.com, gi.akawuku@unizik.edu.ng, so.adejumo@unizik.edu.ng

ABSTRACT

This research investigates the comparative effectiveness of real-time and batch processing approaches in machine learning-driven fraud detection within financial institutions. Fraud detection in finance is a critical and evolving challenge, as fraudulent activities exploit system vulnerabilities and often appear as anomalies within a vast set of legitimate transactions. By implementing classification models K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR) and deep learning models, including Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, this study aims to differentiate and optimize fraud detection performance in both real-time and batch contexts. Real-time processing allows immediate analysis as transactions occur, enabling prompt fraud detection and response, which is essential in scenarios where instantaneous action is needed to prevent losses. Adopting Cross-Industry Standard Processing for Data Mining (CRISP-DM) Methodology leveraging on Google Colab and Python tools. Conversely, batch processing evaluates transactions collectively after a designated interval, providing a more comprehensive analysis by identifying patterns in larger datasets but at the expense of delayed detection. In this study, fraud is characterized by patterns such as unusually high transaction volumes, atypical geographic locations, and irregular transaction timings. Machine learning techniques are employed to analyze these features, distinguishing legitimate from potentially fraudulent transactions. Results indicate strengths and limitations in both processing modes, with real-time offering speed but potential data noise and batch providing accuracy yet delayed detection. This research underscores the value of a tailored approach, integrating machine learning models to enhance fraud detection efficacy and highlights the implications of processing choices for financial institutions aiming to strengthen security frameworks against evolving fraudulent strategies.

Keywords: Anomaly Detection, Batch Processing, Convolutional Neural Networks (CNN), Fraud Detection, Financial Institutions, K-Nearest Neighbors (KNN), Logistic Regression (LR), Long Short-Term Memory (LSTM), Machine Learning, Real-Time Processing.

INTRODUCTION

Financial fraud has emerged as a significant concern for individuals, corporations, and financial institutions worldwide, particularly with the rapid growth of online banking, e-commerce, and digital payment systems. Fraudulent activities, including identity theft, credit card fraud, and money laundering, have led to substantial financial losses, estimated to be in the billions of dollars annually [1]. Traditional fraud detection methods, which often rely on rule-based system and manual reviews, have proven inadequate in keeping pace with the sophisticated tactics employed by fraudsters. The advent of big data and advancements in computational power have paved the way for more effective solutions. Machine learning, a subset of artificial intelligence, enables systems to learn from historical data and identify complex patterns that may indicate fraudulent behavior. Unlike conventional approaches, ML algorithms can adapt to new fraud patterns over time, making them particularly suited for dynamic and evolving financial environments [2]. The application of machine learning in financial fraud detection involves the use of various algorithms, such as supervised learning techniques (e.g., classification algorithms) and unsupervised learning techniques (e.g., clustering and anomaly detection). Supervised learning models are trained on labeled datasets, allowing them to predict fraudulent activities based on historical instances. In contrast, unsupervised learning approaches analyze data without pre-labeled outcomes, identifying unusual patterns that may warrant further investigation [4]. Despite the potential benefits, the integration of machine learning into fraud detection systems presents several challenges. Issues such as data privacy, the need for high-quality labeled

datasets, and the interpretability of complex models must be addressed to enhance trust and usability in financial applications. Furthermore, the fast-paced nature of financial fraud requires continuous model updates and retraining to ensure optimal performance [1].

REVIEW OF RELATED WORKS

The study "Adaptive machine learning models: Concepts for real time financial fraud prevention in dynamic environments" by [2], explores the application of adaptive machine learning models to enhance real-time financial fraud prevention. As financial institutions grapple with sophisticated and ever-evolving fraud schemes, traditional static models fail to adequately respond to these dynamic threats [5]. Adaptive machine learning models address this gap by continuously learning from new data and adapting to emerging fraud patterns [6]. The research highlights advanced techniques like reinforcement learning, online learning, and deep learning, which enable these models to optimize fraud detection, process vast datasets in real time, and identify anomalies with high precision [7]. Additionally, the integration of explainable AI (XAI) ensures transparency and regulatory compliance, fostering trust in these systems. Despite their promise, challenges such as data quality, computational demands, and model interpretability must be addressed to implement these models effectively [8]. Overall, adaptive machine learning provides a robust, responsive, and innovative approach to safeguarding financial systems against fraud while enhancing their reliability and trustworthiness [9].

METHODOLOGY

This research employs a mixed methods approach, combining a quantitative technique, under the Cross Industry Standard Process for Data Mining (CRISP-DM) framework. The quantitative aspect of this study involves collecting and analyzing numerical financial data from a diverse dataset comprising historical and synthetic records of transactions. Key metrics such as transaction frequency, account activity, geographic patterns, transaction volumes, and account balances are used to detect anomalies indicative of fraudulent behavior. Machine learning models—KNN, SVM, LR, CNN, and LSTM—are employed to classify these patterns. Evaluation metrics like precision, recall, F1-score, and accuracy are utilized to compare the performance of real-time and batch processing methods, focusing on the trade-offs between detection speed and accuracy. The study also simulates real-world transaction workflows to assess latency and resource consumption under both processing frameworks. The CRISP-DM framework guides the research through various stages, starting with problem analysis, where the problem domain is defined by reviewing literature and expert opinions. Key characteristics of fraudulent transactions are identified to inform data collection. A large dataset of mobile money transactions was gathered from Kaggle, See dataset, ensuring ethical and privacy compliance. Data preprocessing is performed to handle missing values, noise, and redundancy. The research leverages various tools and technologies, including the Google Colab for programming and code editing, TensorFlow and Scikit-learn for machine learning, Matplotlib for data visualization. By combining quantitative and qualitative methodologies under the structured CRISP-DM framework, this research aims to deliver a robust and practical solution for detecting AI-driven impostor accounts on social media.

RESULTS AND DISCUSSIONS

Dataset Description

The dataset used in this research is the Paysim synthetic dataset, designed to simulate mobile money transactions for fraud detection purposes. This dataset is scaled down to one-fourth of the original dataset described in the research paper *"PaySim: A Financial Mobile Money Simulator for Fraud Detection."* It provides a realistic representation of financial transaction patterns, making it suitable for evaluating fraud detection techniques in both real-time and batch processing contexts.

Each record in the dataset represents a transaction, described by the following fields:

a) **step**: A temporal mapping where each step corresponds to one hour of simulation, allowing for chronological analysis of transaction sequences.

b) **type**: The type of transaction, categorized into five groups: CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER. These types capture the diverse financial activities that could indicate legitimate or fraudulent operations. The figure below shows a graphical representation of the percentage distribution of the categorical values of the "type" field.

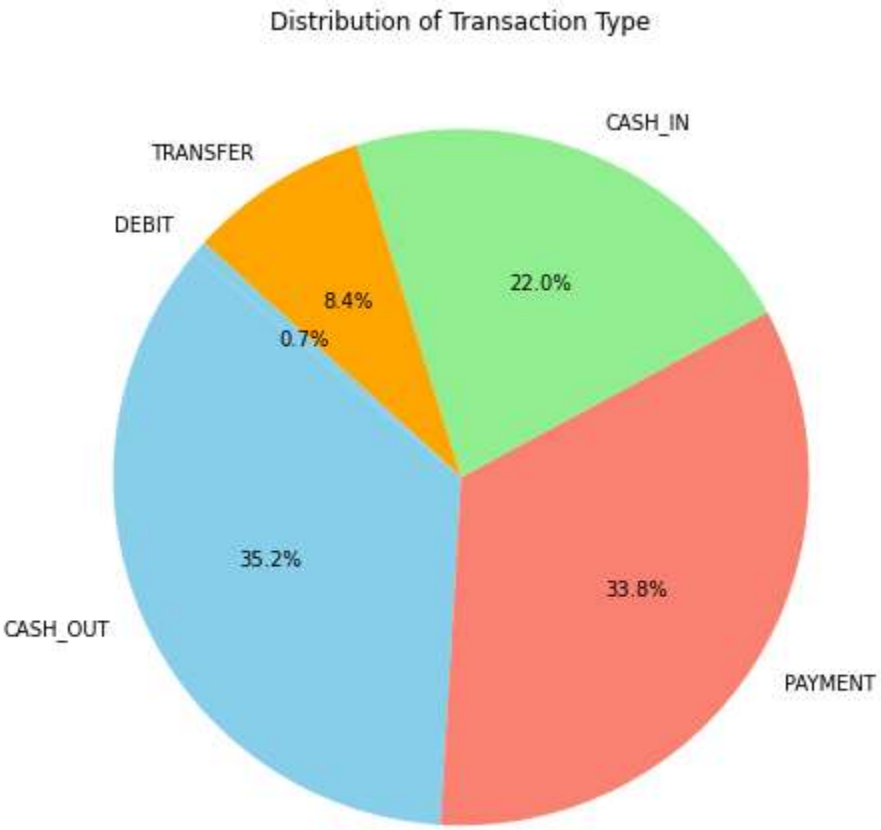

Figure 1: Pie chart representation of the distribution of the different class types

c) **amount**: The transaction amount in the local currency, which provides critical information for identifying suspiciously large or unusual transactions.
d) **Name Orig**: The unique identifier of the customer initiating the transaction, enabling tracking of customer-level activities and behavior.
e) **Old balance Org** and **newbalance Orig**: The account balances of the customer before and after the transaction, respectively, indicating potential inconsistencies or anomalies in financial operations.
f) **Name Dest**: The unique identifier of the transaction's recipient, allowing for the analysis of transfer patterns and recipient networks.
g) **Old balance Dest** and **newbalance Dest**: The recipient's account balances before and after the transaction, useful for detecting abnormal fund inflows or outflows.
h) **isFraud**: A binary field indicating whether the transaction is fraudulent (1) or legitimate (0), providing a ground truth for model training and evaluation.
i) **Is Flagged Fraud**: Flags transactions attempting to transfer amounts exceeding 200,000 in a single transaction, marking them as potential fraud attempts.

This dataset's synthetic nature ensures the absence of sensitive customer data while maintaining the complexity and diversity of real-world transactions. Its granular detail enables robust analysis of transaction behavior, allowing machine learning models to learn patterns indicative of fraud effectively. Furthermore, the dataset's temporal structure and inclusion of diverse transaction types make it ideal for comparing real-time and batch processing approaches to fraud detection.

Data Preprocessing

Imputation of Latent Missing Values

The data has several transactions with zero balances in the destination account both before and after a non-zero amount is transacted [10]. The fraction of such transactions, where zero likely denotes a missing value, is much larger in fraudulent (50%) compared to genuine transactions (0.06%). Since the destination account balances being zero is a strong indicator of fraud, we do not impute the account balance (before the transaction is made) with a statistic or from a

distribution with a subsequent adjustment for the amount transacted. Doing so would mask this indicator of fraud and make fraudulent transactions appear genuine [11]. Instead, below we replace the value of 0 with -1 which will be more useful to a suitable machine-learning (ML) algorithm detecting fraud. The data also has several transactions with zero balances in the originating account both before and after a non-zero amount is transacted [12]. In this case, the fraction of such transactions is much smaller in fraudulent (0.3%) compared to genuine transactions (47%). Once again, from similar reasoning as above, instead of imputing a numerical value we replace the value of 0 with a null value.

Feature-engineering

Motivated by the possibility of zero-balances serving to differentiate between fraudulent and genuine transactions, we take the data-imputation of section 4.2.1 a step further and create 2 new features (columns) namely: **error Balance Orig** and **error Balance Dest**, recording errors in the originating and destination accounts for each transaction respectively. These new features turn out to be important in obtaining the best performance from the ML algorithm that we will finally use.

Data Visualization

The best way of confirming that the data contains enough information so that a ML algorithm can make strong predictions, is to try and directly visualize the differences between fraudulent and genuine transactions. Motivated by this principle, I visualize these differences in several ways in the plots below.

Dispersion over Time

The plot below shows how the fraudulent and genuine transactions yield different fingerprints when their dispersion is viewed over time. It is clear that fraudulent transactions are more homogenously distributed over time compared to genuine transactions. Also apparent is that CASH-OUTs outnumber TRANSFERs in genuine transactions, in contrast to a balanced distribution between them in fraudulent transactions. Note that the width of each 'fingerprint' is set by the 'jitter' parameter in the **plotStrip** function above which attempts to separate out and plot transactions occurring at the same time with different abscissae.

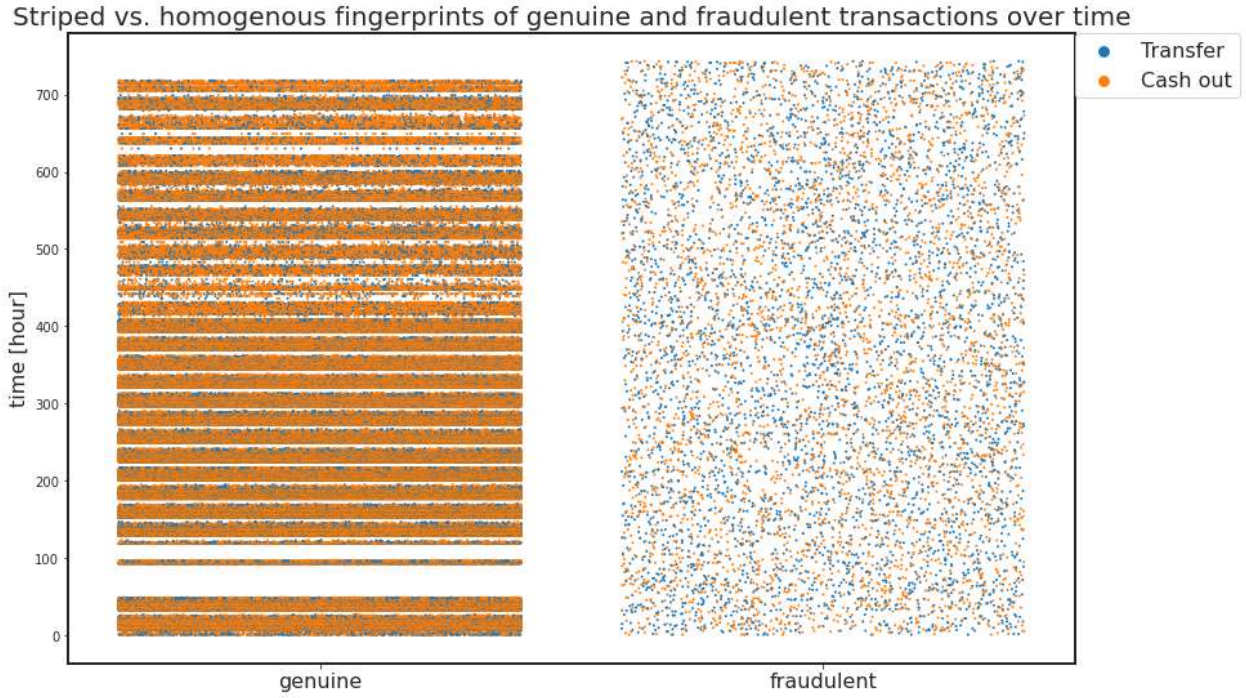

Figure 2: Plot showing how the fraudulent and genuine transactions yield different fingerprints when their dispersion is viewed over time.

Dispersion over amount

The two plots below shows that although the presence of fraud in a transaction can be discerned by the original *amount* feature, the new *errorBalanceDest* feature is more effective at making a distinction.

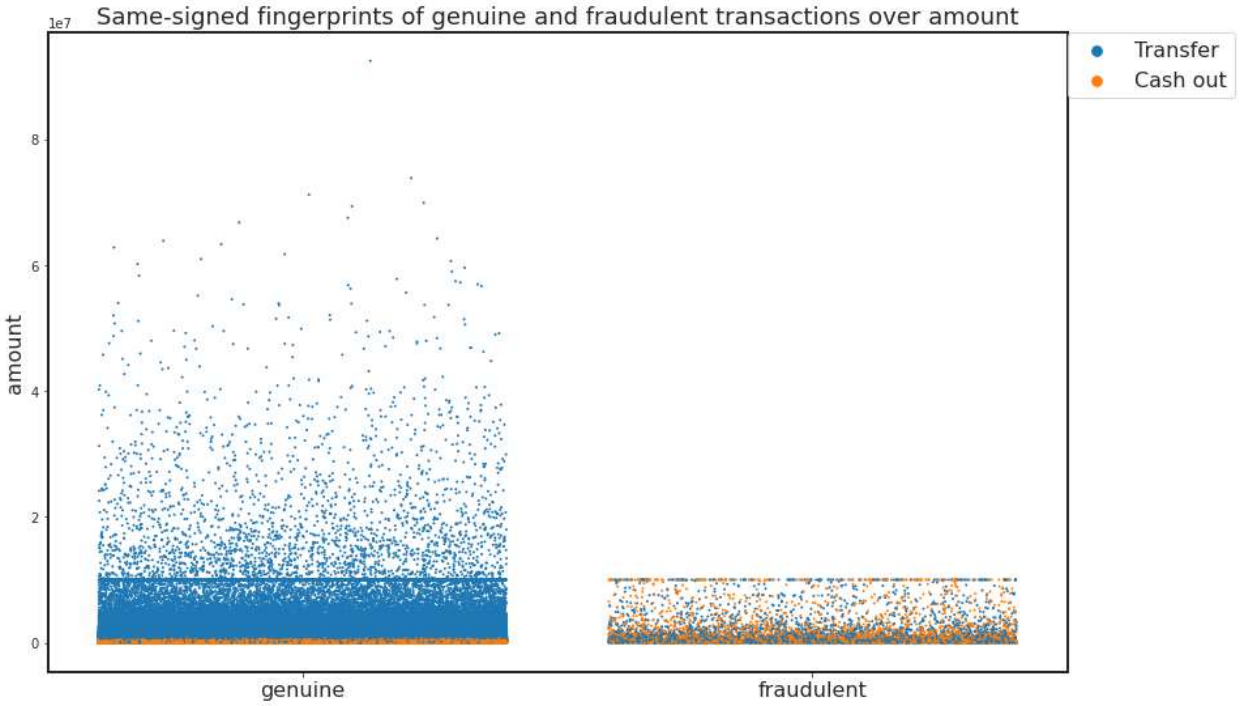

Figure 3: Plot showing the detection of frauds using the amount after transactions.

Dispersion over error in balance in destination accounts

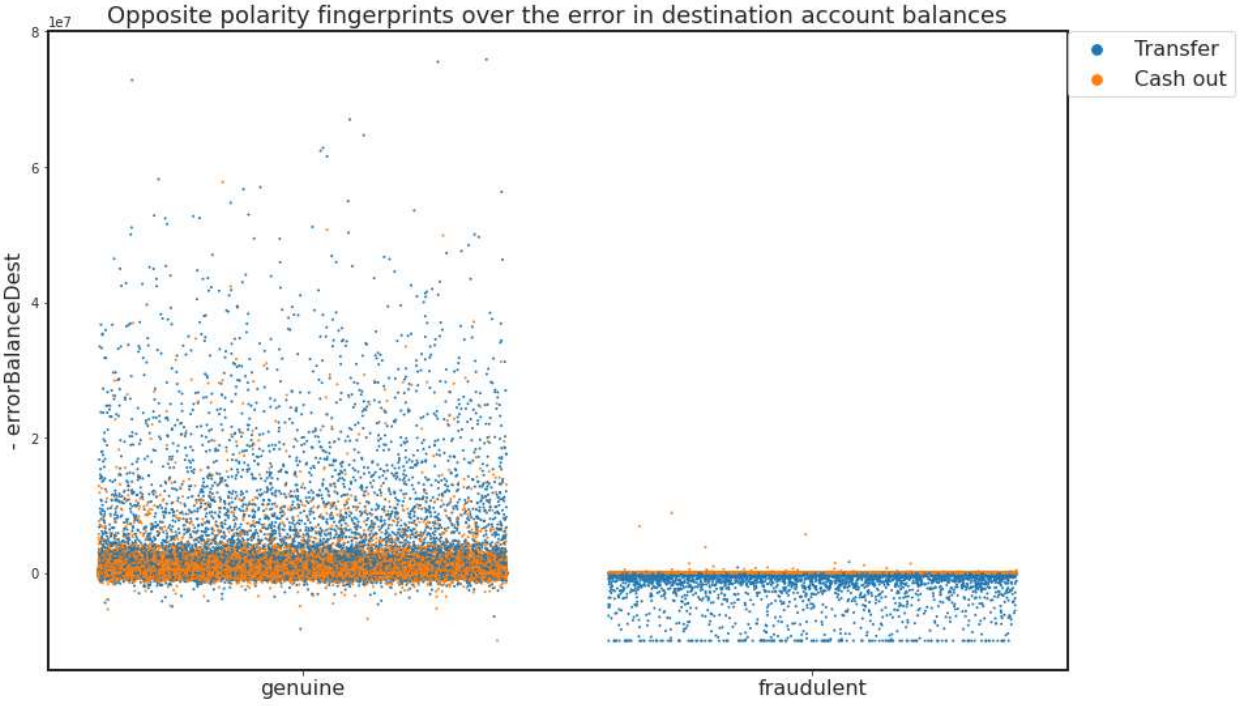

Figure 4: Plot showing the detection of frauds using the amount after transactions.

Separating out genuine from fraudulent transactions

The 3D plot below distinguishes best between fraud and non-fraud data by using both of the engineered error-based features. Clearly, the original *step* feature is ineffective in separating out fraud. Note the striped nature of the genuine data vs time which was anticipated from the figure in section 5.1

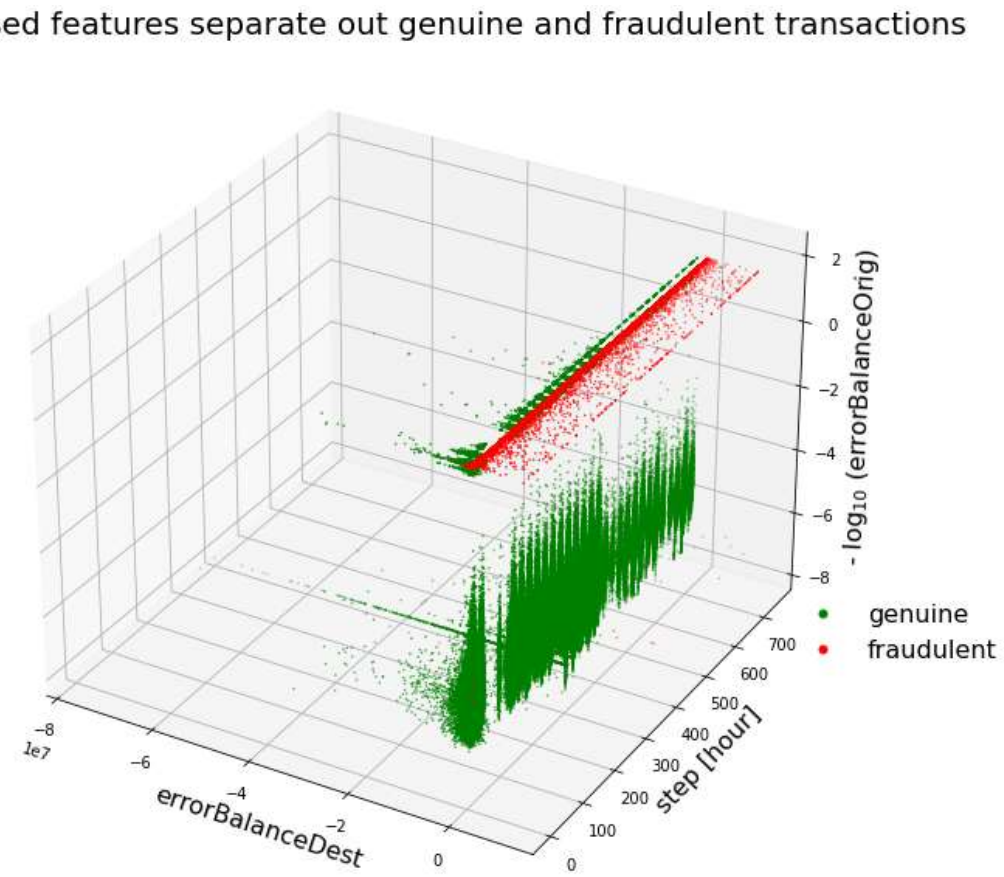

Figure 5: Using the feature-engineered "error Balance Dest" to separate the genuine from fraudulent transactions.

Fingerprints of genuine and fraudulent transactions

Smoking gun and comprehensive evidence embedded in the dataset of the difference between fraudulent and genuine transactions is obtained by examining their respective correlations in the heatmaps below.

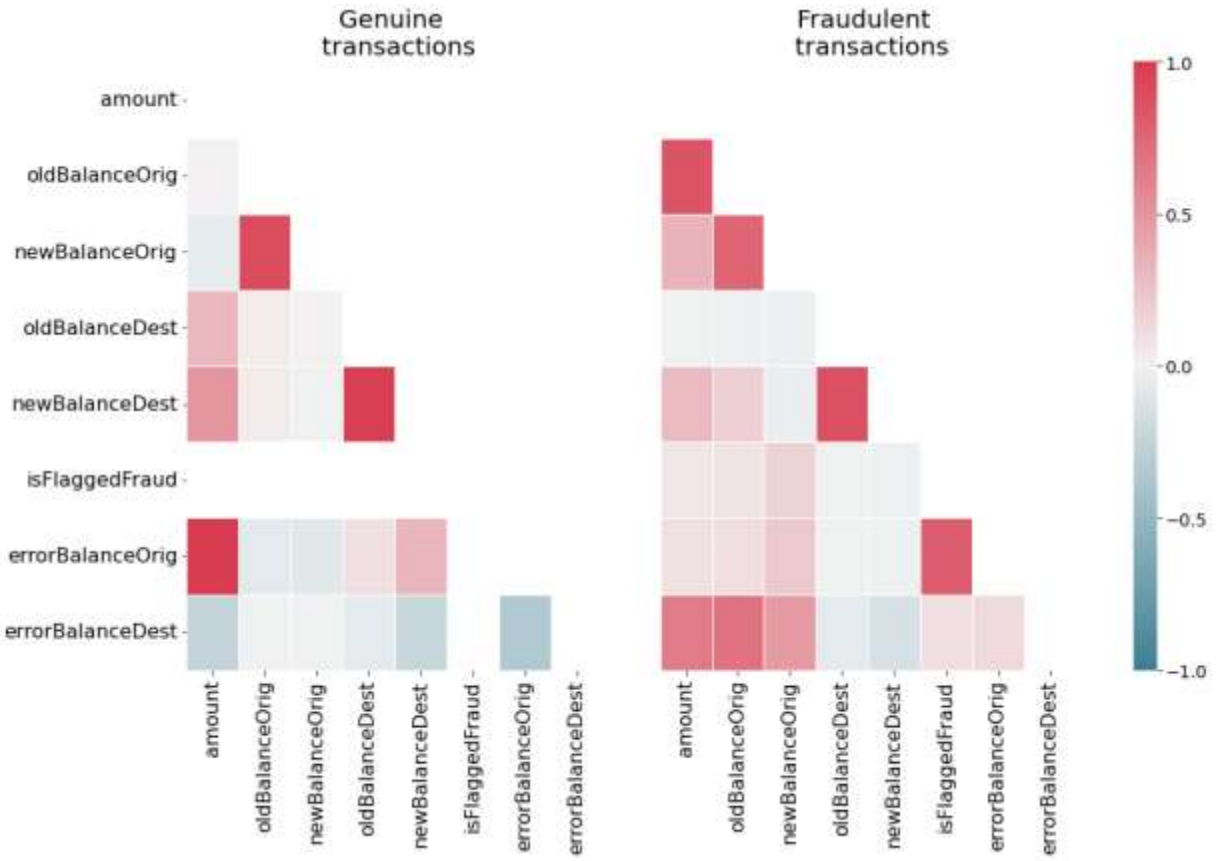

Figure 6: Heatmap showing the correlation among the features in both Genuine and Fraudulent transactions.

Machine Learning to Detect Fraud in Skewed Data

Having obtained evidence from the plots above that the data now contains features that make fraudulent transactions clearly detectable, the remaining obstacle for training a robust ML model is the highly imbalanced nature of the data.

a) ***Selection of metric***: Since the data is highly skewed, the Area Under the Precision-Recall Curve (AUPRC) was used rather than the conventional Area Under the Receiver Operating Characteristic (AUROC). This is because the AUPRC is more sensitive to differences between algorithms and their parameter settings rather than the AUROC [3].

b) ***Selection of ML algorithm***: A first approach to deal with imbalanced data is to balance it by discarding the majority class before applying an ML algorithm. The disadvantage of undersampling is that a model trained in this way will not perform well on real-world skewed test data since almost all the information was discarded [13]. A better approach might be to oversample the minority class, say by the synthetic minority oversampling technique (SMOTE) contained in the 'imblearn' library [14]. Motivated by this, I tried a variety of anomaly-detection and supervised learning approaches. I find, however, that the best result is obtained on the original dataset by using a ML algorithm based on ensembles of decision trees that intrinsically performs well on imbalanced data. Such algorithms not only allow for constructing a model that can cope with the missing values in our data, but they naturally allow for speedup via parallel-processing. Among these algorithms, the extreme gradient-boosted (XGBoost) algorithm used below slightly outperforms random-forest. Finally, XGBoost, like several other ML algorithms, allows for weighting the positive class more compared to the negative class --- a setting that also allows to account for the skew in the data [15].

What are the important features for the ML model?

The figure below shows that the new feature **errorBalanceOrig** that we created is the most relevant feature for the model. The features are ordered based on the number of samples affected by splits on those features.

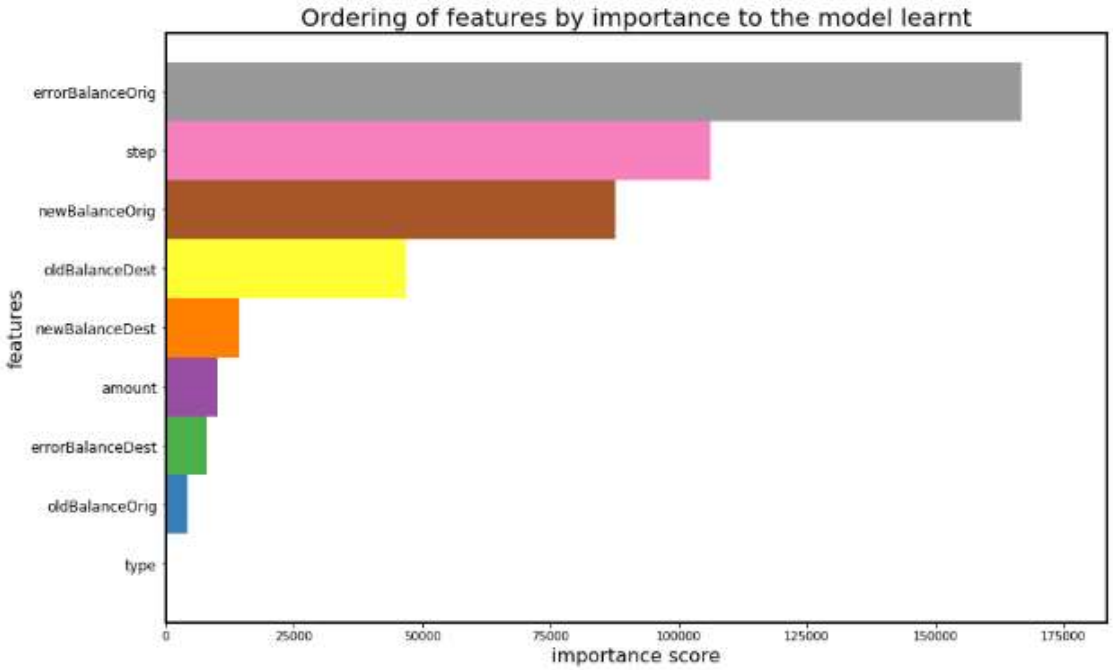

Figure 7: Plot showing the ordering of features by importance to the model.

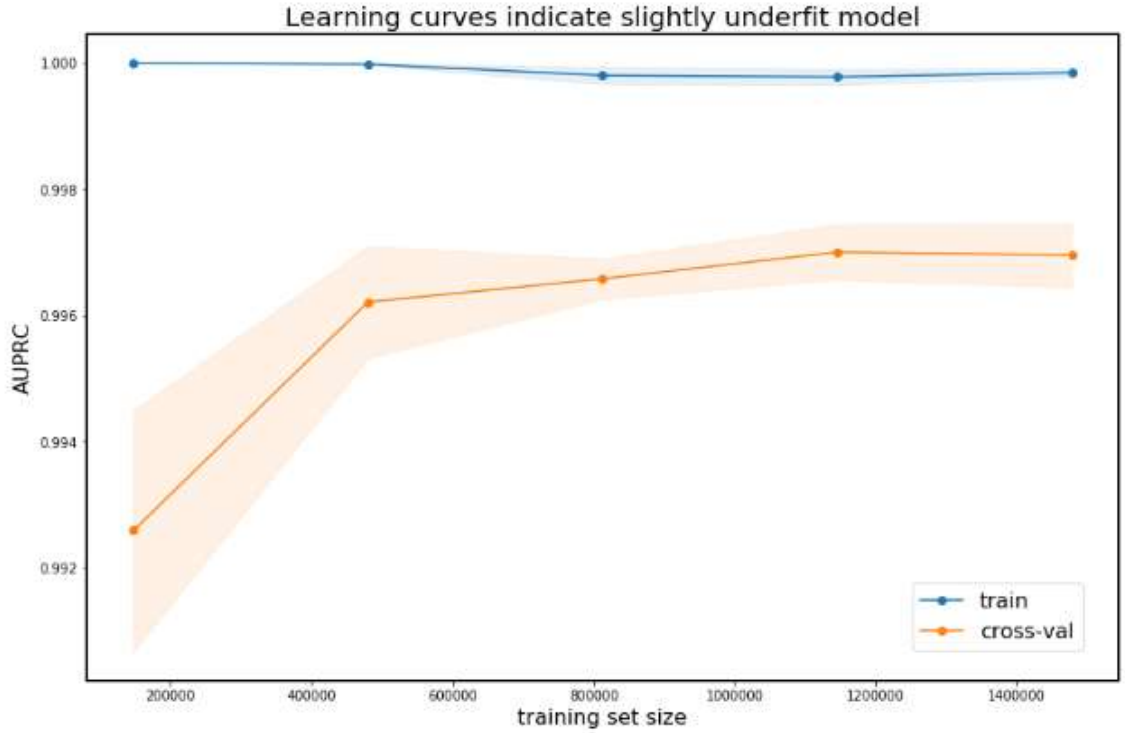

Figure 8: Learning curve indicating a slightly underfit model.

Real-Time and Batch Processing: Accuracy and Performance Tradeoffs

Our analysis of fraud detection systems revealed distinct tradeoffs between real-time and batch processing approaches in terms of accuracy and performance, each offering unique strengths and limitations depending on operational requirements. Real-time processing analyzes transactions as they occur, offering immediate detection and response to potential fraud. This approach is critical in scenarios where timely intervention can prevent financial losses. During the study, models implemented for real-time analysis, such as CNN and LSTM, excelled in processing continuous streams of data with minimal latency. However, the results highlighted that real-time systems are prone to higher false-positive

rates. This tradeoff stems from the limited contextual information available at the moment of analysis, which can lead to misclassification of legitimate transactions as fraudulent. Real-time processing also demands significant computational resources, especially when handling large volumes of concurrent transactions.

Table 1: Performance Metrics Evaluation for Fraud Detection Using Real Time Processing

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.57	0.6	0.58
Support Vector Machine	0.81	0.84	0.71	0.75
K-Nearest Neighbors	0.65	0.62	0.56	0.59
CNN + LSTM	0.93	0.91	0.88	0.89

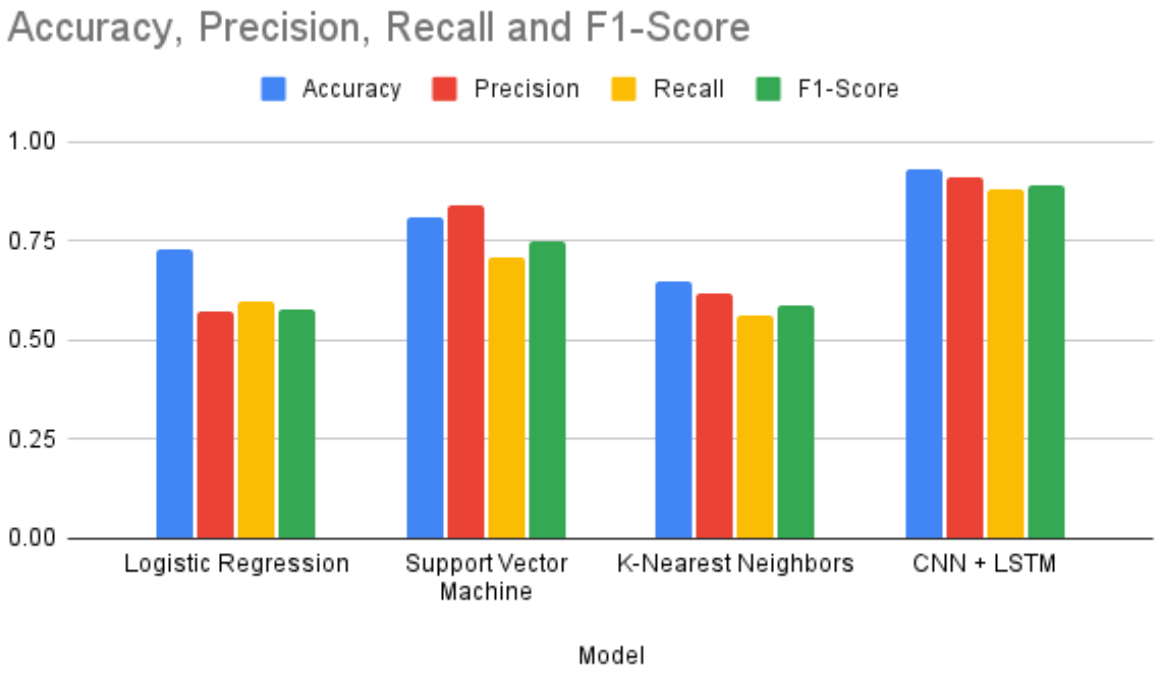

Figure 9: Evaluation of the models used in Real Time processing.

Batch processing evaluates transactions in aggregated intervals, allowing for more comprehensive analysis. This method benefits from access to larger datasets, enabling models like Logistic Regression and Support Vector Machines to identify subtle patterns and correlations. Our findings indicated that batch processing achieved higher accuracy compared to real-time processing, particularly in detecting complex fraud schemes. However, the delayed response time inherent in batch processing limits its effectiveness in preventing fraud during high-stakes, time-sensitive transactions. This delay is a notable tradeoff for scenarios requiring immediate action.

Table 2: Performance Metrics Evaluation for Fraud Detection Using Batch Processing

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.74	0.69	0.63	0.66
Support Vector Machine	0.81	0.84	0.74	0.78
K-Nearest Neighbors	0.71	0.67	0.63	0.65
CNN + LSTM	0.98	0.94	0.91	0.92

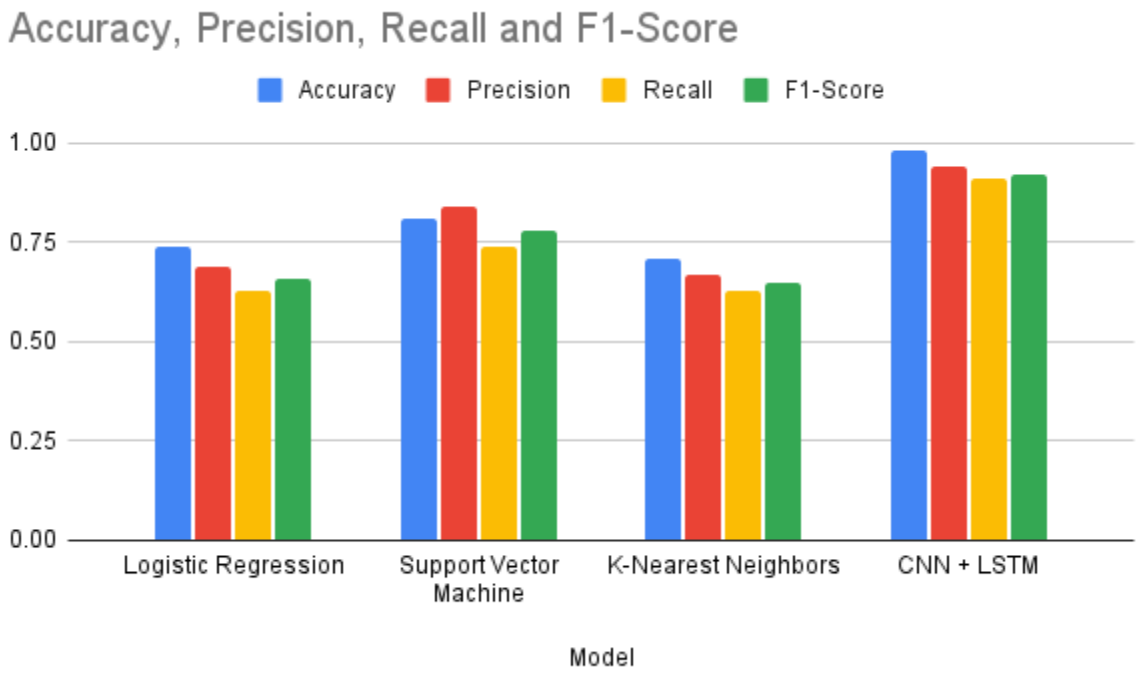

Figure 9: Evaluation of the models used in Batch processing.

The choice between **Real-time and Batch processing** depends on the operational priorities of financial institutions. Real-time processing favors immediacy and responsiveness, while batch processing offers improved accuracy through deeper analysis. A hybrid approach, leveraging the strengths of both methods, may present an optimal solution for robust fraud detection systems.

CONCLUSION

This research explored the effectiveness of machine learning techniques for fraud detection using the Paysim synthetic mobile money transaction dataset. Through careful data preprocessing, including strategic handling of missing values and innovative feature engineering (notably the creation of errorBalanceOrig and errorBalanceDest), the study enhanced the dataset's ability to distinguish between fraudulent and genuine transactions. Data visualization revealed distinctive behavioral patterns between fraudulent and non-fraudulent activities, reinforcing the dataset's utility for model training. Fraudulent transactions demonstrated more uniform dispersion over time and unique balance error signatures, which were further confirmed by correlation heatmaps and 3D visualizations. To address the significant class imbalance in the dataset, ensemble-based machine learning models—particularly XGBoost—were employed. These models not only managed skewed distributions effectively but also handled missing or engineered values without compromising performance. The errorBalanceOrig feature emerged as the most influential factor in fraud classification. The study further evaluated the trade-offs between **real-time** and **batch processing** methods. Real-time systems, while providing immediate responses critical in fast-paced environments, suffered from higher false-positive rates due to limited contextual data. On the other hand, batch processing achieved superior accuracy and recall by leveraging more comprehensive datasets, although at the cost of delayed detection. Performance metrics underscored these differences, with CNN+LSTM models performing best in both contexts achieving up to **0.98 accuracy** in batch processing and **0.93** in real-time scenarios. This highlights the potential of deep learning models in extracting temporal and contextual patterns from transaction data. In summary, this research demonstrates that effective fraud detection in mobile money transactions can be achieved through a combination of:

- Rigorous preprocessing and feature engineering,
- Appropriate model selection tailored for imbalanced data,
- Strategic deployment of real-time and batch processing based on operational needs.

Future directions could explore hybrid frameworks that dynamically switch between real-time and batch modes based on risk thresholds or transaction characteristics, thereby combining the strengths of both approaches for a more adaptive and resilient fraud detection system.

REFERENCES

1. Potter, K. and Axel Egon, A. (2024). MACHINE LEARNING IN FINANCIAL FRAUDDETECTION. Financial Management. See Article.
2. Bello, H., Ige, A, and Ameyaw, M. (2024). Adaptive machine learning models: Concepts for real time financial fraud prevention in dynamic environments. World Journal of Advanced Engineering Technology and Sciences. 12(2):021-034. See Article.
3. Atadoga, J.O., Nembe, J.K., Mhlongo, N.Z., Ajayi-Nifise, A.O., Olubusola, O., Daraojimba, A.I. and Oguejiofor, B.B., (2024). Cross-Border Tax Challenges and Solutions In Global Finance. Finance & Accounting Research Journal,6(2), pp.252-261
4. Manoharan G, Dharmaraj A, Sheela SC, Naidu K, Chavva M, Chaudhary JK. Machine learning-based real-time fraud detection in financial transactions. In2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) 2024 May 9 (pp. 1-6). IEEE.
5. Ahmmed MJ, Rahman MM, Das AC, Das P, Pervin T, Afrin S, Tisha SA, Hassan MM, Rahman N. COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR BANKING FRAUD DETECTION: A STUDY ON PERFORMANCE, PRECISION, AND REAL-TIME APPLICATION. American Research Index Library. 2024 Nov 15:31-44.
6. Bello HO, Ige AB, Ameyaw MN. Deep learning in high-frequency trading: conceptual challenges and solutions for real-time fraud detection. World Journal of Advanced Engineering Technology and Sciences.2024;12(02):035-46.
7. Preciado Martínez PM, Reier Forradellas RF, Garay Gallastegui LM, Náñez Alonso SL. Comparative analysis of machine learning models for the detection of fraudulent banking transactions. Cogent Business & Management. 2025 Dec 12;12(1):2474209.
8. Ayub MI, Bhattacharjee B, Akter P, Uddin MN, Gharami AK, Islam MI, Suhan SI, Khan MS, Chambugong L. Deep Learning for Real-Time Fraud Detection: Enhancing Credit Card Security in Banking Systems. The American Journal of Engineering and Technology. 2025 Apr 30;7(04):141-50.
9. Rehan H. Leveraging AI and cloud computing for Real-Time fraud detection in financial systems. Journal of Science & Technology. 2021;2(5):127.
10. Li F, Chen Z. Dynamic quantification anti-fraud machine learning model for real-time transaction fraud detection in banking. Discover Computing. 2025 May 1;28(1):59.
11. Alonge EO, Eyo-Udo NL, Ubanadu BC, Daraojimba AI, Balogun ED, Ogunsola KO. Enhancing data security with machine learning: A study on fraud detection algorithms. Journal of Data Security and Fraud Prevention. 2021 Jan;7(2):105-18.
12. Hanae AB, Abdellah BE, Saida EL, Youssef GA. End-to-end real-time architecture for fraud detection in online digital transactions. International Journal of Advanced Computer Science and Applications. 2023;14(6).
13. Hanbali N, El-Yahyaoui A. Advanced machine learning and deep learning approaches for fraud detection in mobile money transactions. Innovations in Systems and Software Engineering. 2025 Apr 18:1-21.
14. Alghofaili Y, Albattah A, Rassam MA. A financial fraud detection model based on LSTM deep learning technique. Journal of Applied Security Research. 2020 Oct 1;15(4):498-516.
15. Poudel S, Movinuddin, Gutta S, Kommu RK, Upadhyay J, Hasan MN, Poudel K. Credit Card Batch Processing in Banking System. InInternational Conference on Advances in Computing Research 2024 Mar 29 (pp. 83-96). Cham: Springer Nature Switzerland.

CITE AS: Egwu C. Victor, Akawuku I. Godspower and Adejumo O. Samuel. Comparative Study of Real-Time and Batch Processing Approaches in Machine Learning-Based Fraud Detection for Financial Institutions. IAA Journal of Scientific Research 12(2):60-70. https://doi.org/10.59298/IAAJSR/2025/1226070.00