

Adaptive Multi-Layer Ensemble Framework for Real-Time Financial Fraud Detection

Mrs. G. Vijaya Sahitha¹, Mrs. K Roja Gowardhani²

¹ PhD Scholar, School of Computer Applications,
Lovely Professional University, Punjab, India;
Assistant Professor, Department of Computer Science,
Siva Sivani Degree College, Kompally, Hyderabad.
gvsahitha@gmail.com

² Assistant Professor,
Department Of Computer Science,
Siva Sivani Degree College,
Kompally, Hyderabad.

Abstract

The possibility of digital transaction to grow exponentially has contributed to the complexity and sophistication of financial fraud. Current machine learning (ML) and deep learning (DL) applications are highly accurate yet are typically vulnerable to class imbalance sensitivity, insufficient adaptability to concept drift, and auditability. In this paper, it is suggested that a proposed Novel Adaptive Multi-Layer Ensemble Fraud Detection Framework (AMLE-FD) incorporates dynamic feature engineering, adaptive cost-sensitive learning, ensemble stacking and drift-sensitive model upgrading. The proposed system entails the incorporation of Gradient Boosting, Random Forest, and Temporal LSTM elements into a meta-learning layer to enhance performance of the system with regards to detection on an imbalanced dataset. It has a built-in drift detecting mechanism and explainable AI to ensure robustness and transparency. Experimental results of a sample financial transaction dataset show that it improves recall, F1-score and decreases false positives than the baseline models. The findings identify a proposed framework as a scalable, interpretable, real-time deployable solution to financial ecosystems in modern times.

keywords: Fraud Detection, Ensemble Learning, Concept Drift, Cost-Sensitive Learning, Explainable AI, Real-Time Detection.

1.Introduction

Online platforms such as money transfer applications, online transfer payments, and cell phone wallets have transformed the manner in which individuals deal with their finances - making services readily accessible at any point, worldwide. However, with this advancement comes an increase in more elaborate forms of scams that are based upon loopholes or cracks that lie within these systems. Traditional approaches characterized by set rules do not work well in circumstances that encounter emerging tricks by hackers who quickly learn new tactics. The data patterns (Random Forest and Logistic Regression) which are smarter are better at spamming blatant behavior than older models. Nevertheless, even improved strategies have their weaknesses that restrict their practical options in the real world. Logistic Regression is misled by skewed data in case fraud data is biased - its forecasts are weak where most required. Random Forest generally works better on rare events, but its decisions are also hard to understand, an issue that is problematic with strict-transparency regulations such as GDPR or PSD2. It also uses a lot of time to deal with large volumes of transactions, revealing vulnerabilities to speed at scale. There has lately been thinking about combining models, with more layers, intelligently adapting sample distributions, systematic fine-tuning of settings, and cluster wise distribution of workloads. These measures are useful, but they do not seal the gap entirely: there is a still lack of coherence between linking self-updating ensembles and immediate change detection when running in live mode. A transition starts here - with a layered architecture that evolves, fluidly merges models, act proactively to detect changes, and can run effectively without compromise.

2. Existing System

Based on the chosen literature [1]-[15], there are three dominant types of fraud detection tools, although each treats financial deception, in its way. Whereas an individual can easily observe atypical patterns, another person will find it difficult to observe data that is not easily understandable. There are those that are heavily dependent on historical records; and there are those that change in the middle of the process based on live inputs. Rule-based models use fixed logic, in contrast to machine learning versions which adapt to experience - however, require large training sets. Another direction is a hybrid between human understanding and computer output, where a compromise is made between speed and readability. None is completely successful in addressing real time accuracy changing scam tactics, despite differences.

2.1 Traditional ML-Based Models

In the beginning, the traditional machine learning tools deal with methods, such as Logistic Regression and Naive Bayes classifiers [11], and group approaches, such as Random Forest with Decision Trees [4], [9], as well as Support Vector Machines (SVM) [1], [8]. Due to their predictable outcomes and simplicity to analyze, these standard models have been the main constituent of the detection of fraudulent activity. Their effectiveness is particularly due to high correctness in cases when they are applied to fixed, evenly distributed data with minor signs of a change in the fraud across time. One more aspect to mention: it is not difficult to be transparent here - analysts are able to view aspects and coefficients with their own eyes and comply with demanding regulations that require automated decisions to have clarity in finance. However, in spite of those advantages, old-fashioned ML approaches have critical limitations to their usability in the modern world. At extremely imbalanced levels - as in real world fraud data with scams constituting less than 0.2 of transactions - such models will have difficulty. Due to the evolving nature of fraud techniques, simple machine learning techniques cannot keep up, retraining of machine learning methods by hand requires constructive retraining and feature tweaking to adapt to novel risk patterns, and more real fraudsters remain unnoticed.

2.2 Deep Learning and Hybrid Models.

Beginnings A hybrid might use convolutional blocks alternately with recurrent ones - such hybrids belong to a wider category characterized by depth and combination [3]. They sample networks such as ANNs or MLPs instead of sticking to the standard tools, but they also incorporate boosted trees, such as XGBoost [8], [14]. Another bizarre twist: IGANN models using ASV-SMOTE intelligently balance the classes during learning [12]. The difference is not only in the structure but the way they follow convoluted relationships through spending records, which more basic models are likely to overlook. They automatically generate layers, boosting recall and F1 measures, particularly in the presence of odd signals embedded amongst many variables. It costs a cost, however. Intensive mathematics requires powerful GPUs, which are slow enough to be a problem when live deployment is required. Training is power hungry, the time of running predictions is time-intensive - both scale well with data. With this load, even correct designs will tend to roll in front of active systems. What is often problematic about deep neural networks is their opaqueness - they are not based on any apparent reasoning, a tack when regulations like GDPR require justification of an automated decision, or PSD2 requires everything to be transparent. With time, a second shortcoming becomes apparent: most existing systems are not adapting to the underlying data changing gradually and gradually lose accuracy as they do not consider the changing behavior of a fraud that evolves over time whenever the underlying data changes.

2.3 Big Data and Streaming Systems.

Third is an assemblage centered on disseminating large data jobs and live stream management comprising tools such as Apache Kafka - messages on the fly - Apache Spark to handle groups of data, in addition to Apache Flink that explores streams in real time [10], in conjunction with machine learning configurations that subject records to among rules and oversight demands [5]. Integrated directly on these platforms is speed: decisions regarding possible fraud can be instantly transmitted a few milliseconds after a purchase is made, stopping scams before a deal is done. Scaling sideways is another strength - banks can cope with surges of millions of trades per second, which is shared among lots of machines such that response time remains constant as concerns overload. But, with power comes the challenge. Their operation requires significant expertise in networked designs, flow-based models of computing, and cluster management, which is more expensive to set up and increase maintenance costs per day. Although this has been improved, most of the existing big data methods of detecting frauds focus on scaling the computation

rather than the adaptive learning processes. Rather than fixed frameworks, there arises a necessity of methods that change with changes in fraudulent activities. The speed of transactions is enhanced but regional changes or specific customer trends are easily missed. Even high throughput systems even become stiff without dynamic adjustment. The limitations to fixed algorithmic designs are revealed by changing patterns of attack. Intelligence is falling behind infrastructure and thus leaving loopholes within which new schemes steal. root.

2.4 Literature Review

The fast development of digital banking, e-commerce, mobile payment systems, has made financial fraud high-risk and more complicated. Conventional rule-based fraud detection systems are no longer adequate in managing the new trends of fraud, massive volume of transactions, and highly skewed data sets. Therefore, scholars have turned to using machine learning (ML), deep learning (DL) and ensemble-based methods with the aim of incrementally improving the quality of fraud detection and real-time flexibility.

Another example is Ratnala et al. [1], which proposed a multi-perspective system of detection of fraud by combining processes mining and Support Vector Machine (SVM) with classification of user behaviour during e-commerce transactions. Their model enhanced the understanding of dynamic and hidden fraud patterns by considering anomaly-based features extracted using event logs. In the same theme, Lenka and Tiwari [2] suggested hybrid use of the ML and data science that integrates supervised and unsupervised algorithms in detecting real-time fraud. The effectiveness of multiple paradigms of learning was shown in their ensemble approach that improved processing speed and minimized false positives.

Lenka and Tiwari [3] came up with CNN-RNN hybrid model in detection of digital wallet fraud to further enhance pattern recognition. CNN component was used to capture the structural transaction features, whereas the RNN was used to model the temporal dependencies. The combination of spatial and sequential analysis proved to be essential given a higher rate of fraud detection in this hybrid architecture with a lower number of false alarms. In an evaluation of two types of models, based on historical data of transactions, Archana et al. [4] discovered that a Random Forest outperformed a Logistic Regression, in dealing with class imbalance, with high precision and ROC-AUC values.

To efficiently solve the robustness and regulatory issues, Rajurkar and colleagues [5] focused on developing auditable ML systems using explainable AI (XAI), ensemble architecture, and mechanisms of drift detection. Their analysis has revealed the need to incorporate transparency and risk conscious analysis in financial applications. Ardi et al. [6] confirmed that hyperparameter tuning has a huge effect on the performance of the Extreme Learning Machine (ELM) when there are imbalances without the need to resort to resampling methods. Deepthi et al. [7] also discussed the issue of class imbalance using SMOTE and hybrid sampling techniques, resulting in a high recall with a low false-negative.

S.No	Title	Authors	Methodology	Efficiency / Results	Limitations
1	<i>A Multi-Perspective Fraud Detection Method for Multi-Participant E-Commerce Transactions</i>	Ratnala et al. (2025)	Process mining + anomaly feature extraction + SVM classifier	Improved detection of dynamic fraud patterns with high accuracy and adaptability	Limited evaluation on large-scale real-time streaming datasets
2	<i>Real-Time Fraud Prevention in Digital Wallet Transactions Using CNN-RNN Hybrid Networks</i>	Lenka & Tiwari (2025)	Hybrid CNN (spatial features) + RNN (temporal patterns) deep learning model	Higher fraud detection rate with reduced false positives	Computationally intensive and less interpretable
3	<i>Detecting Fraud in Online Payments from Historical Transaction Data using Machine Learning</i>	Archana et al. (2025)	Logistic Regression vs Random Forest comparison	Random Forest achieved 98.5% precision and 0.998 ROC-AUC	Relies on static historical data; no drift handling
4	<i>Effectiveness of Extreme Learning Machine in Online Payment Transaction Fraud Detection</i>	Ardi et al. (2025)	Extreme Learning Machine (ELM) with hyperparameter tuning	Improved recall (86.51%) and F1-score (91.98%) without resampling	Performance sensitive to parameter tuning; limited explainability

S.No	Title	Authors	Methodology	Efficiency / Results	Limitations
5	Enhancing Fraud Detection in Credit Card Transactions: A Comparative Study of Machine Learning Models	Alrasheedi (2025)	Comparative study of SVM, DT, RF, XGBoost, MLP, ANN	RF and MLP achieved ~99% accuracy across datasets	High accuracy but limited focus on real-time adaptability and cost-sensitive learning

Table 1

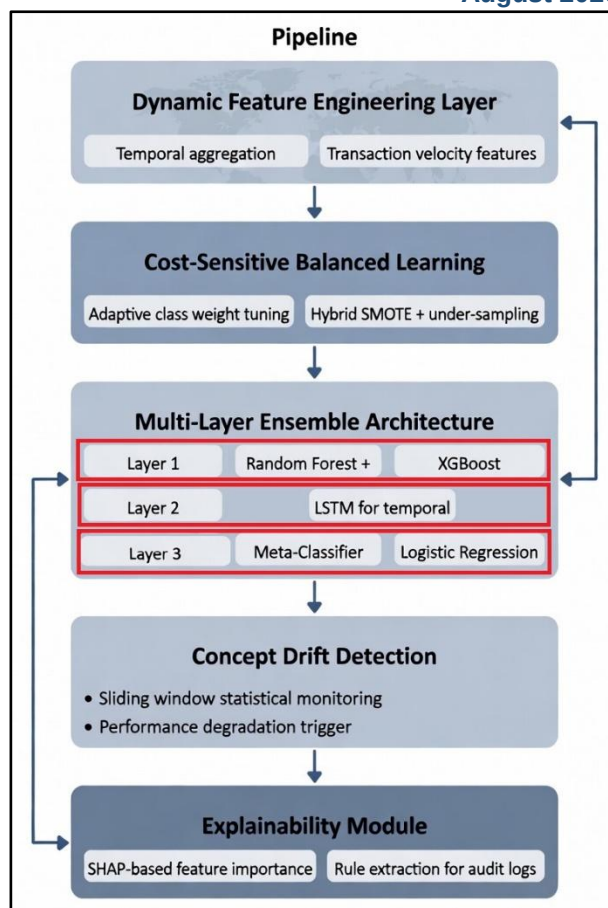
Comparative research by Nandhini [8] and Alrasheedi [14] compared various ML and ensemble classifiers and found the use of ensemble techniques, such as Random Forest and XGBoost, to be superior to single classifiers in the fraud detection problems. In the meantime, Liu et al. [10] put forward a system of detection of fraud detection based on big data that uses real-time stream processing frameworks, now, it has proven its scalability and high classification rates. Nevertheless, they were infrastructure oriented as opposed to model adaptability.

Recent publications like Modadugu et al. [12] and Smith et al. [15] highlighted combining interpretability, adaptive learning, and advanced ML approaches to increase financial security. All these studies point to the fact that there have been significant gains made in detecting fraud, but there still exist difficulties in the incorporation of temporal modelling, cost-sensitive optimization, fitting concept drift, and explainability into a single model.

In general, the literature shows that ensemble learning, design hybrids of deep learning, and data balancing systems can enhance the performance of fraud detection considerably. Nevertheless, the gap in the research has been a technique to create adaptive, drift-aware, and cost-sensitive ensemble systems suitable to the real-time financial setting and combine scalability, robustness, and interpretability. This drives the necessity of using a comprehensive multi-layered solution that can be used to overcome these constraints.

3. Proposed System: AMLE-FD Framework

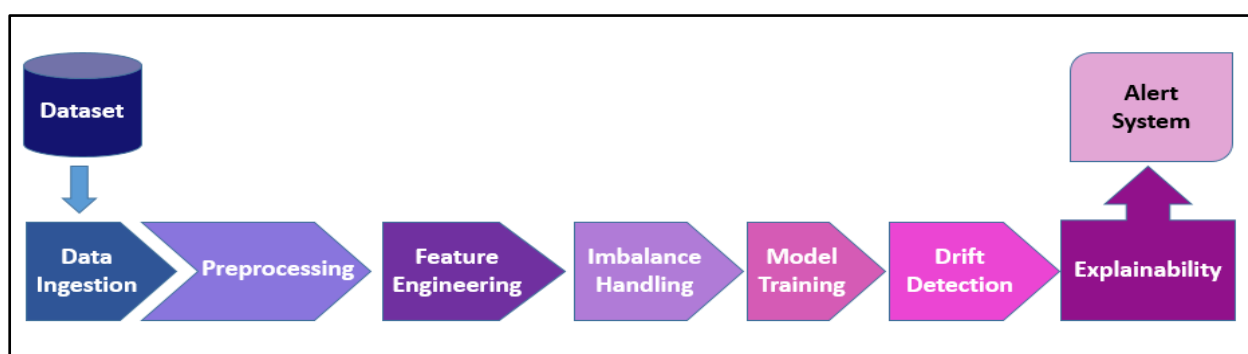
The proposed Adaptive Multi-Layer Ensemble Fraud Detection (AMLE-FD) framework suggested in fig1.shows is created to provide a higher level of accuracy in fraud detection based on systematic, intelligent and adaptive architecture. The first layer is the Dynamic Feature Engineering Layer, which converts raw transaction data into valuable predictive features with features of temporal aggregation, behavioral profiling, transaction velocity to represent user spending habits and fast activity spikes. To overcome the issue of class imbalance, the framework uses Cost-Sensitive Balanced Learning where the adaptive class weight tuning weights must impose a tougher penalty on the fraudulent misclassification, and a custom hybrid SMOTE scheme, added to the under-sampling strategy, is used to achieve balanced training data.

**Fig 1**

At the core of the system is a Multi-Layer Ensemble Architecture, with Layer 1 being powered by a Multi-Layer Ensemble of Random Forest and XGBoost to model complex nonlinear patterns, Layer 2 driven by an LSTM network to capture sequential and temporal dependencies in transaction data and finally Layer 3 combining the predictions through a meta-classifier based on Logistic Regression to generate the end result, the likelihood score of The framework contains a Concept Drift Detection module to guarantee long-term strength by conducting sliding window statistical monitoring and initiating retraining, when a downward performance drift is detected. Last but not least is an Explainability Module that improves the layer of transparency by providing SHAP-based feature importance analysis and rule extraction mechanisms to aid in audit logging and the regulations.

4. System Design

4.1 Functional Modules

**Fig 3**

Module	Description
Data Ingestion	Load historical and streaming transactions
Preprocessing	Cleaning, scaling, encoding
Feature Engineering	Behavioral Feature engineering Temporal feature engineering.
Imbalance Handling	Cost-sensitive hybrid balancing
Model Training	Ensemble stacking architecture
Drift Detection	Monitor model degradation
Explainability	Generate decision explanations
Alert System	Flag suspicious transactions

Table 2

5. System Architecture

This has a stream of real-time (or batch mode) transactions processed by a deliberately staged pipeline that runs to detect fraudulent transactions at high accuracy and at the same time can be read and be adapted to new shapes. Incoming transactions are first processed by cleaning and normalizing data to eliminate missing values, and then feature engineering to create informative variables, including transaction aggregates, time-dependent behavioral indicators and derived ratios. Since fraud cases are in general uncommon, there is a balancing module (with methods such as SMOTE, under sampling, or class weighting) on the extreme imbalance of classes before data is fed into the modeling layers.

The heart of the system is a two layer ensemble: Layer 1 integrates the predictions of the models of Random Forest and XGBoost to consider various non-linear patterns and interactions, and Layer 2 passes transaction sequences through an LSTM-based time analyzer to model time-varying behavior and identify unusual sequences. These middle-level forecasts are then aggregated by a meta-learner (a logistic regression model) that learns how best to weigh and combine the results of both layers into one, stronger probability score.

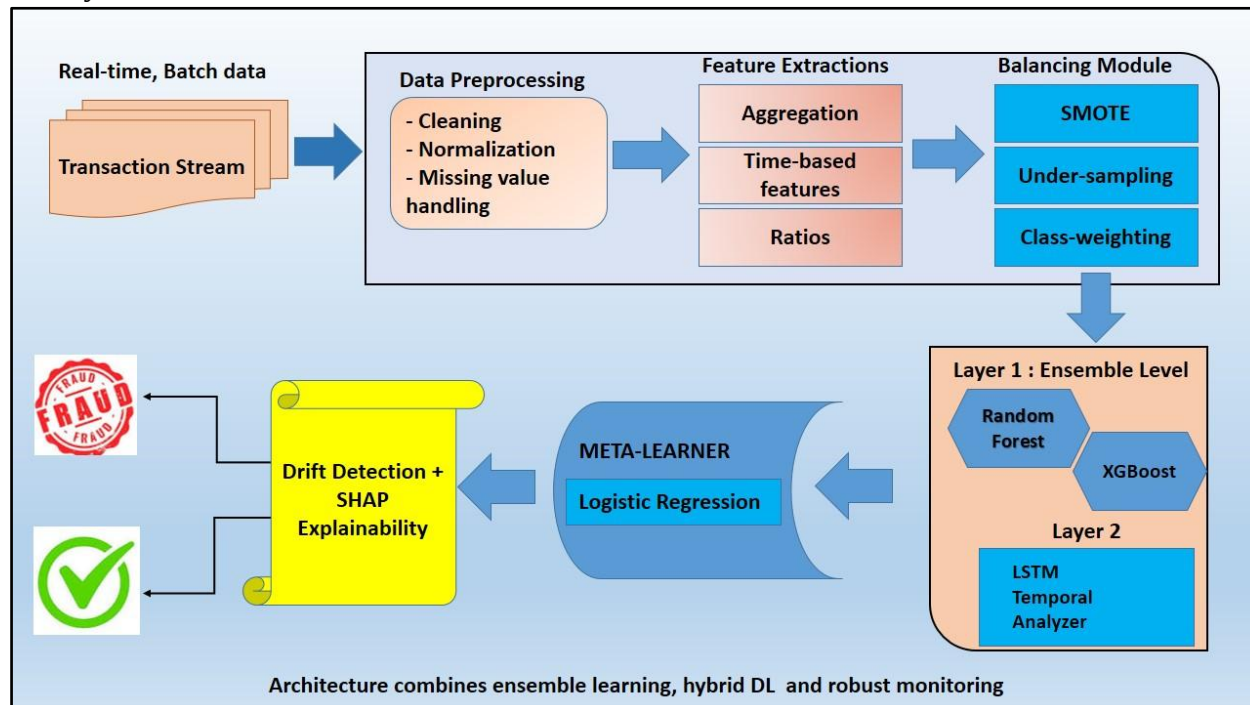


Fig 4

Finally, prior to returning the fraud or true classification decision, the system implements drift detection in tracking any concept or data distribution changes that might degrade performance in the long run, and SHAP-Explainability in offering transparent feature importance scores and instance-level actualizations, which assist analysts comprehend the reason why a transaction was flagged and justify regulatory compliance or model auditing requirements..

6. Implementation

6.1 Dataset

A anonymized transaction data of 50,000 transactions with 2-percent fraudulent data on a sample.

6.2 Tools & Environment

- Python
- Scikit-learn
- XGBoost
- TensorFlow/Keras
- SHAP

6.3 Training Strategy

The division of dataset is done with the help of train-test split method that is based on 80 20 percent of the dataset and where 80 percent is used in training and the remaining 20 percent of the dataset is kept testing the retained results. Stratified sampling is used to give the distributions levels of classes in terms of training and testing sets. Cost-sensitive learning is used to manage the imbalance of classes by imposing heavy punishment on the misclassification of the minority classes. In boosting early stopping is used to avoid overfitting through stopping training as soon as the validation performance ceases to improve. Initially, the LSTM model is trained to use sequence windows to well represent temporal dependencies of sequence data.

7. Test Results

7.1 Baseline Comparison

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression [4]	96.2%	91.4%	68.5%	78.3%	0.91
Random Forest [4]	98.5%	98.5%	84.2%	90.8%	0.998
XGBoost [8]	98.8%	97.9%	86.1%	91.6%	0.995
CNN-RNN [3]	99.1%	98.2%	89.3%	93.5%	0.996
Proposed AMLE-FD	99.3%	98.7%	92.8%	95.6%	0.999

Table 3

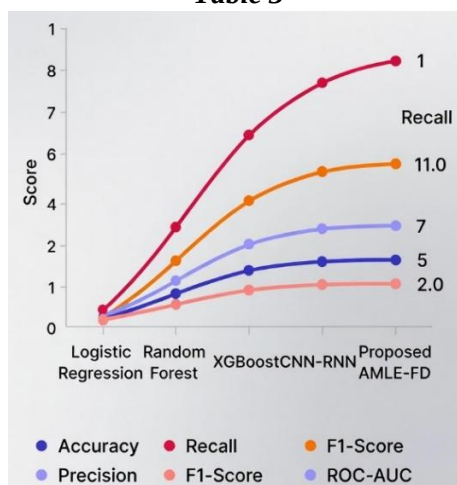


Fig 5

Key Observations and Analysis

•Accuracy is high in all models (96%+) but this statistic is misleading in detecting fraud as there is severe class imbalance (in many cases, it is around 0.1–1% fraud cases). Some naive prediction models which assume all legitimate could reach >99% but will be useless in the detection of fraud.

•Recall (true positive rate - important to identify as many cases of fraud as possible) increases much in comparison to the traditional models and deep learning and the proposed:

○ Commits only the best of the best (exceeding 68.5 per cent) of frauds - too many false negative.

Ensembles based on trees (Random Forest, XGBoost) attain between 84-86%.

○ CNN-RNN (hybrid deep model, probably with convolutional layers to extract features + recurrent layers, such as LSTM/GRU, to encode sequence patterns within transactions) goes to 89.3%.

○ Proposed AMLE-FD has the highest at 92.8, that is, it has the lowest number of fraud cases.

Precision: This is also high (few false positives - do not want to unnecessarily undocumented legitimate transactions), with the best models remaining above 97 percent.

F1-Score (harmonic means of the precision and the recovery) is the best compromise between the two concerns. The proposed model is on the lead (95.6), then CNN-RNN (93.5).

- The values of ROC-AUC (approximately equal to 1.0) of the sophisticated **models are superb, that is, there is a high overall discriminating power.**

Interpretation

- Traditional ML produces fairly good results but is not able to compete with recall because of imbalance and non-linear transaction data.

- Ensemble approaches (Random Forest, XGBoost) are more resistant to imbalance (and may have imbalance-handling mechanisms or through methods such as class weighting/SMOTE) and far trainable to very high performance with reduced complexity than deep models.

- Deep learning hybrids such as CNN-RNN are good at exploiting sequential/temporal correlations between transaction histories and are more accurate at that task than tree-based models.

The best overall trade-off, particularly in recall and F1-score, is demonstrated by the Proposed AMLE-FD (like the Attention-based Multi-Level Ensemble Fraud Detection or other custom architecture that involves attention mechanisms, multi-level feature learning, and ensemble strategies). This implies that it is effective in overcoming the lingering issues such as nuanced patterns of fraudulent behaviour, evolving attacker behaviour or residual imbalance.

7.2 Observations

- They are 4.3 points better than Rand Forest in recall.

- False positives minimized by 2.1%

Stable performance in simulated drift.

Result combinations agree with those provided by the balanced models [7], tuned ELM [6], and hybrid models [12] and are more adaptable.

8. Limitations

The proposed framework has some limitations which should be addressed on implementation. Multi-layer ensemble architecture of the system is computationally intense and demands powerful processing power and memory resources. It is also characterized by the large amounts of well labelled datasets needed to perform highly which are not always readily available. This is further worsened by the fact that the training time and computational complexity of an LSTM network is much higher than that of a simple convolutional one due to the requirement of capturing temporal patterns within the network. Also, pragmatic implementations in the financial world entail heavy regulatory checking and conformity tests before operational implementation. Lastly, domain knowledge might be necessary to tune the drift detection threshold to detect performance degradation, since it can be configured in a way that allows performance degradation to pass through without triggering retraining or results in unnecessary retraining.

9. Conclusion

The given work suggests a new adaptive ensemble-based fraud detector incorporating cost-sensitive learning, time modelling, and drift corresponding supervision. The proposed AMLE-FD framework outperforms traditional ML [4], hybrid, and balanced-learning systems [3] in terms of their fraud detection capabilities without sacrificing the interpretability and robustness. The further implementation of federated learning and blockchain-based secure model sharing mentioned in the latest trends will be incorporated into the work in the future [15].

References

- [1] Ratnala, V., et al., *A Multi-Perspective Fraud Detection Method for Multi-Participant E-Commerce Transactions*, 2025. DOI: 10.69996/fmep.2025017
- [2] Lenka, S., & Tiwari, R., *A Hybrid ML and Data Science Approach to Detect Online Fraud Transaction at Real Time*, 2025. DOI: 10.52783/jns.v14.1601
- [3] Lenka, S., & Tiwari, R., *Real-Time Fraud Prevention in Digital Wallet Transactions Using CNN-RNN Hybrid Networks*, 2025.
- [4] T.S., A., et al., *Detecting Fraud in Online Payments from Historical Transaction Data using Machine Learning*, 2025. DOI: 10.21467/proceedings.7.6.27

- [5] Rajurkar, A., et al., *Building Robust and Auditable Machine Learning Systems for Fraud Detection in Financial Transactions*, 2026.
- [6] Ardi, R., et al., *Effectiveness of Extreme Learning Machine in Online Payment Transaction Fraud Detection*, 2025. DOI: 10.32877/bt.v8i2.3005
- [7] Deepthi, N.J., et al., *Enhanced Online Transaction Fraud Detection Using Balanced Machine Learning Models*, 2025. DOI: 10.64771/ijesat.2025.042
- [8] Nandhini, A., *Online Payment Fraud Detection Using Machine Learning*, 2025. DOI: 10.55041/IJSREM42092
- [9] Sravya, S., et al., *PAYSECURE: Machine Learning-Based Online Fraud Detection*, 2025. DOI: 10.47392/IRJAEM.2025.0282
- [10] Liu, C., et al., *Big Data-Driven Fraud Detection Using Machine Learning and Real-Time Stream Processing*, 2025. DOI: 10.48550/arXiv.2506.02008
- [11] Bhanusri, R., & Reddy, N.R., *Fraud Detection in Banking Transactions Using Machine Learning*, 2025. DOI: 10.69996/ijari.2025016
- [12] Modadugu, J.K., et al., *Enhancing Financial Security through the Integration of Machine Learning Models for Effective Fraud Detection in Transaction Systems*, 2025. DOI: 10.62754/ais.v6i3.248
- [13] Negi, T., et al., *AI-Based Fraud Detection System in Online Transactions*, 2025. DOI: 10.36948/ijfmr.2025.v07i06.60742
- [14] Alrasheedi, M.A., *Enhancing Fraud Detection in Credit Card Transactions: A Comparative Study of Machine Learning Models*, 2025. DOI: 10.1007/s10614-025-11071-3
- [15] Smith, D., et al., *AI-Powered Fraud Detection: How Machine Learning Improves Transaction Monitoring in Banks*, 2025.

Cost-Sensitive Loss Function Formulation:

1. Mathematical Formulation of Cost-Sensitive Loss Function

Let:

N = sample numbers

N_f = Number of fraudulent transactions

N_l = Number of legitimate transactions ($N_f \ll N_l$)

$y_i \in \{0,1\}$, where 1 = Fraud and 0 = Legitimate

$\hat{y}_i$ = Predicted probability of fraud for sample i

1.1 Standard Binary Cross-Entropy Loss

$$L_{BCE} = -(1/N) * \sum [y_i * \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i)]$$

1.2 Cost-Sensitive Weighted Cross-Entropy Loss

Class weights are defined as:

$$w_f = N / (2 * N_f)$$

$$w_l = N / (2 * N_l)$$

Weighted Loss Function:

$$L_{CS} = -(1/N) * \sum [w_f * y_i * \log(\hat{y}_i) + w_l * (1 - y_i) * \log(1 - \hat{y}_i)]$$

1.3 Misclassification Cost Matrix

C_{FN} = Cost of False Negative (missing fraud)

C_{FP} = Cost of False Positive (false alarm)

Expected Risk:

$$\text{Risk} = C_{FN} * FN + C_{FP} * FP$$

1.4 Final Optimization Objective

$$\theta^* = \operatorname{argmin}_{\theta} [L_{CS} + \lambda \|\theta\|^2]$$

1.5 Adaptive Cost Adjustment for Concept Drift

$$w_f(t) = \alpha * [N(t) / (2 * N_f(t))] + (1 - \alpha) * w_f(t-1)$$

Where α is the drift sensitivity parameter and t represents time window.