International Journal of Safety and Security Engineering
Vol. 16, No. 10, October, 2026, pp. 2235-2246
Journal homepage: http://iieta.org/journals/ijsse

Drift-Aware Incremental Learning for Dynamic Fraud Detection in Payment Systems Using LightGBM and Passive Aggressive

Hadi M. R. Al Lawati[1*], Mohammad A. N. Al-Azawi[2], Anazida Zainal[1], Bander Ali Saleh Al-Rimy[3], Muhammad Zafran bin Muhammad Zaly Shah[1]

[1] Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor Bahru 81310, Malaysia
[2] Computing Sciences Department, Gulf College, Muscat 133, Oman
[3] PAIDS Research Center, School of Computing, University of Portsmouth, Portsmouth PO1 3HE, United Kingdom

Corresponding Author Email: m.alazawi@gulfcollege.edu.om

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.161004

Received: 13 August 2026
Revised: 30 September 2026
Accepted: 9 October 2026
Available online: 31 October 2026

Keywords:
fraud detection, concept drift, incremental learning, online learning, payment systems

ABSTRACT

Fraud in payment cards results in billions of dollars in losses, and fraudulent patterns continue to evolve, causing static machine learning (ML) models to become outdated. While studies focus on data imbalances and concept drift detection, few address continuous, low-overhead model adaptation without full retraining. This paper proposes a drift-aware incremental learning framework for real-time fraud detection in Payment card transactions. Phase 1 leverages a hierarchical pipeline that utilises Mutual Information (MI) with SelectKBest for feature selection and adaptive synthetic sampling (ADASYN) for class balancing. Phase 2 combines a Convolutional Neural Network (CNN) confidence gate with a modified Early Drift Detection Method (EDDM)-based detector and Adaptive Windowing (ADWIN) to identify detector-attributed candidate-drift signals. The innovative contribution is in Phase 3, where a weighted ensemble of a frozen Light Gradient Boosting Machine (LightGBM) classifier and Passive Aggressive (PA) incremental learner updates only from detector-identified candidate-drift misclassified samples, enabling online adaptation without extensive retraining. This model is evaluated on four datasets, which are a real bank dataset from one GCC bank, the Kaggle European Credit Card dataset, BankSim, and the ESPAM. The nested fold evaluation method (Temporal Forward Chaining) and the stratified random evaluation method applied to the ESPAM Dataset yielded the following results: A perfect accuracy of 1.000, along with a F1-Score of 0.940 when evaluated against the Real Dataset; an accuracy of 0.998, along with a F1-Score of 0.606 when evaluated against the Credit Card Dataset; an accuracy of 0.994, along with a F1-Score of 0.829 when evaluated against the BankSim Dataset; and finally, an accuracy of 0.916, along with a F1-Score of 0.902 when evaluated against the ESPAM Dataset. The model demonstrated strong and stable performance across all datasets and conditions, although the sample size of valid chronologically ordered folds was too small for formal statistical testing. These results demonstrate the performance of the proposed drift-aware incremental learning framework under the evaluated fold-based conditions.

1. INTRODUCTION

One of the most serious threats to the global banking community is financial fraud. Gross fraud losses on card-based payments amounted to 28.65 billion in 2019, as defined by the Nilson Report, equivalent to 6.8¢ of gross fraud losses for every $100 of transaction volume [1]. Fraudulent online payments will cost merchants an estimated $343 billion from 2023 through 2027 [2]. Card-Not-Present (CNP) fraud has been the most common type of payment card fraud in the United Kingdom since 2014 [3].

However, traditional methods of fraud detection are typically based on rules. Rule-based methods are relatively simple to create and implement; however, these types of methods have inherent limitations. Rule-based methods do not allow for automated updates of features based on changing fraud patterns. Instead, rule-based methods require human development and maintenance of features [4]. Machine learning (ML) and deep learning (DL) models have given better pattern recognition than static rule-based systems. However, there is a major and frequently under-treated issue: concept drift, wherein data distribution changes statistically with time, which leads even well-trained models to gradually degrade detection accuracy as fraud patterns change [5].

The prior framework [6] used an Early Drift Detection Method-Adaptive Windowing-Convolutional Neural Network (EDDM-ADWIN-CNN) classifier framework, able to predict gradual and abrupt concept drift of payment card streams, obtaining a performance reaching up to 99.99%. Although it obviously works well for drift detection, it does not propose

guidance for the important problem of how the model can be continuously amended and effectively adapted to detector-identified drift without having to perform complete periodic retraining, which is computationally prohibitive for high-frequency transactional environments.

This paper advances to the third stage by building a weighted set consisting of the Light Gradient Boosting Machine (LightGBM) classifier and the Passive Aggressive (PA) incremental learner. The PA learner updates the model only on drift-flagged misclassified samples, offering targeted online adaptation. This three-stage pipeline is rigorously validated to ensure generalisability across four datasets covering both financial and non-financial domains. Figure 1 illustrates the comprehensive architecture.

The proposed adaptive model is organised according to three main research objectives. The first solution includes a hierarchical preprocessing pipeline to produce class-balanced, low-noise feature sets that are statistically optimal across diverse datasets. Second, it develops a confidence-gated dual drift detector designed to identify candidate signals associated with gradual and abrupt changes while limiting unnecessary detector activations. Third, it develops a drift-triggered incremental ensemble that can continuously adapt to detector-identified fraud-pattern changes without requiring full model retraining, thereby addressing the catastrophic forgetting problem that limits current incremental techniques. The rest of this paper is structured as follows. Related work is reviewed in Section 2. Section 3 presents datasets and methodology in detail. Section 4 presents the results and discusses the findings. Section 5 concludes and identifies future directions.

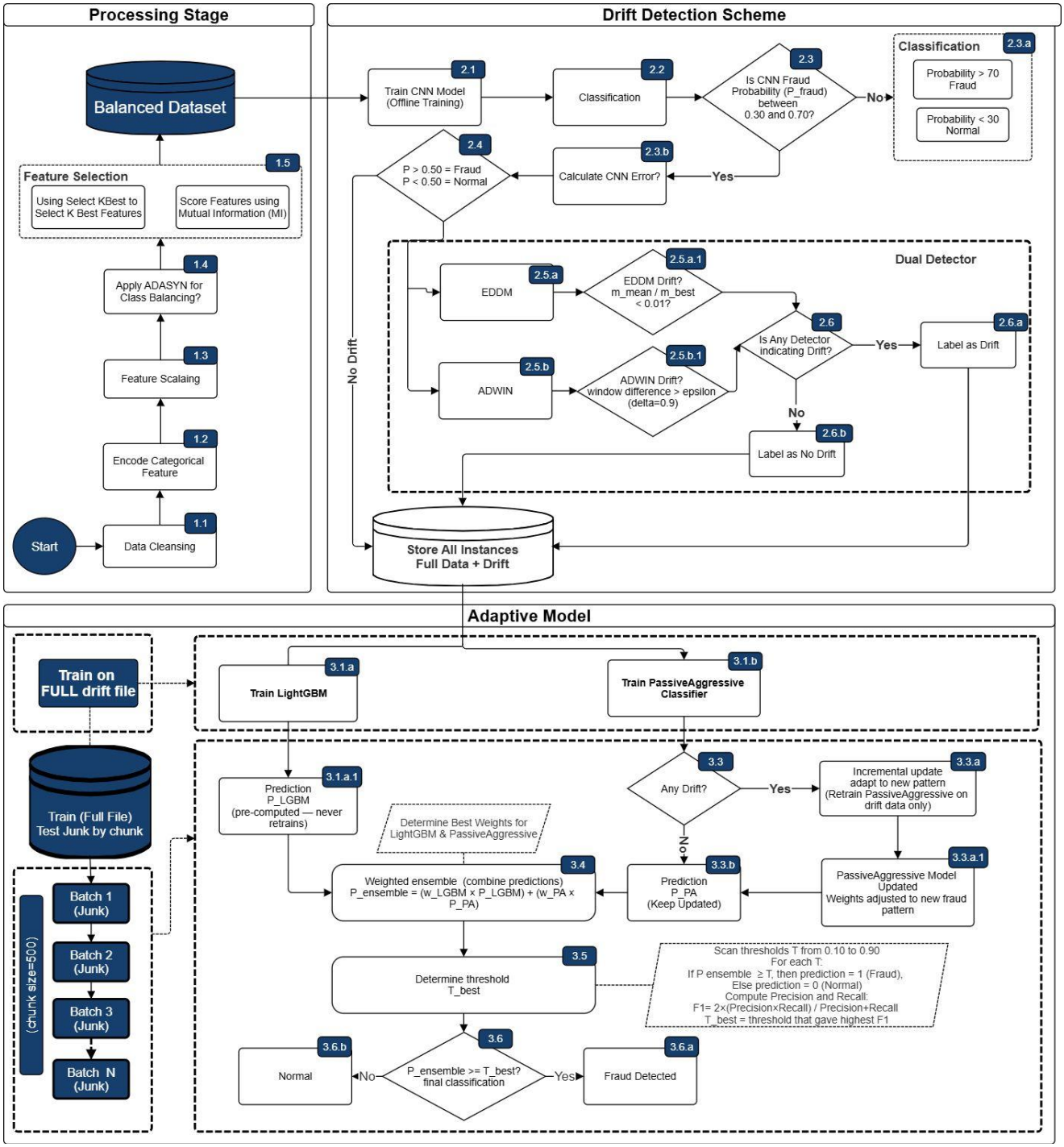


Figure 1. Three-stage drift-aware incremental learning adaptive model

2. RELATED WORK

2.1 Feature selection and class imbalance

Resampling-based approaches combine traditional and ensemble classifiers with SMOTE-based or undersampling methods [7-10]. The methods examined include logistic regression, random forest, XGBoost, J48, SVM, and K-means [7, 8]. Reported accuracies range from 93.8% to 98.5%, with AUC values reaching 99.37% [7, 8, 10]. Most approaches primarily address class imbalance without accounting for feature selection or concept drift. Although information gain has been used to select 18 features, the resulting model does not capture nonlinear relationships among features or incorporate a mechanism for adapting to concept drift [11]. Practical limitations also remain. Additional user involvement in fraud assessment may limit scalability when processing large volumes of banking transactions [9], while bagging and boosting increase computational complexity [10]. Overall, these studies treat class imbalance as a separate preprocessing problem rather than integrating it with feature selection and adaptation to changing fraud patterns. These limitations motivate Stage 1 of the proposed framework.

2.2 Concept drift detection

Concept drift refers to changes in the statistical properties of the target distribution caused by evolving fraud patterns, changes in customer behaviour, or shifts in the payment environment [5]. If left unaddressed, concept drift increases false alarms [12], degrades the performance of static models [13], and creates a need for adaptive mechanisms to maintain model accuracy over time [14]. Several methods have been proposed to detect drift, but adaptation following detection remains limited. An approach combining adaptive synthetic sampling (ADASYN) and ADWIN with a DNN-based model performs well in detecting abrupt drift but requires model retraining after detection because incremental learning is not supported [15]. Other approaches combine ADWIN with a chimp-optimised MLP [16] or use KL divergence and Cramér's V with ensemble classifiers [17]. These methods identify changes in data distribution but do not continuously update the classification model following drift detection [16, 17]. Furthermore, the ensemble-based approach may introduce computational overhead for real-time processing [17]. Overall, the reviewed approaches focus on drift detection, while adaptation is either omitted or handled separately. The proposed framework addresses this limitation by combining drift detection in Stage 2 with incremental model adaptation in Stage 3.

2.3 Incremental learning

Previous studies distinguish between active and passive adaptation in incremental learning. Active approaches retrain the model when drift is detected, whereas passive approaches continuously update the model using techniques such as sliding windows or ensembles [5]. However, both approaches may face difficulties in handling gradual or recurring concept drift. Incremental learning has demonstrated better performance than batch retraining in credit card fraud detection, although catastrophic forgetting remains an important limitation [18]. Incremental gradient boosting trees have also outperformed static batch models, but they require labelled data at fixed intervals and lack an explicit drift-detection mechanism [19]. LightGBM has been widely used for its efficiency and predictive performance on large, imbalanced datasets [20], while the PA classifier supports efficient online updates as new labelled samples become available. However, these techniques are typically studied separately, without combining a stable base model with an online learner that is updated only after concept drift is detected. Three main research gaps can be identified from these limitations:

•Preprocessing, drift detection, and incremental adaptation are seldom integrated into a unified framework;

•Incremental learning without a declared drift detection system leads to avoidable updates and catastrophic forgetting;

•No prior work has effectively connected a frozen LightGBM ensemble to a PA incremental learner utilising a confidence-gated dual drift detector.

All three gaps are addressed within the proposed framework.

3. METHODOLOGY AND PROPOSED FRAMEWORK

Four datasets reflecting distinct operational contexts were selected for evaluation. Table 1 summarises their key characteristics.

No published ground-truth drift locations exist for the Credit Card dataset; all drift statistics reported for this dataset are inferred from the proposed EDDM/ADWIN detectors' own output and cannot be independently validated against ground truth. Transaction counts reflect original dataset totals prior to preprocessing. Post-preprocessing detector-identified candidate-drift sample counts are presented in Section 4.2.

Table 1. Dataset specifications

Dataset	Source	Year	Features	Transactions	Fraud/Spam	Drift
Credit Card	Kaggle [21]	2013	31	284,807	492 (0.17%)	Not labelled
Real Dataset	One GCC Bank (Private)	2019	200	66,523	120 (0.18%)	Yes
BankSim	Synthetic simulator [22]	–	Mixed	1,170,767	High volume	Yes
ESPAM	UCI Repository [23]	1999	57	4,601	1,813 spam	Yes

3.1 Stage 1—Hierarchical data preprocessing

Preprocessing has three steps, each dealing with a different data quality issue. Experimental results show that feature selection prior to ADASYN balancing performs better than the reverse order, as selecting features on the original class distribution prevents biasing feature rankings toward synthetic minority class samples:

•Data Cleansing: NULL values, duplicate records, and zero-variance columns are removed. Categorical features are encoded using LabelEncoder for ordinal and OneHotEncoder for nominal variables; numerical missing values are imputed by column medians.

•Feature Selection (MI + SelectKBest): Mutual Information

quantifies statistical dependence between each feature and the target class, independent of linear assumptions. Optimal feature configurations were determined empirically for each dataset based on the MI ranking and SelectKBest evaluation.

•Class Balancing (ADASYN): ADASYN generates synthetic minority-class instances proportionally to local learning difficulty, concentrating generation on the classification-hardest regions of feature space. This produces a more informative, balanced dataset with lower overfitting risk than uniform SMOTE replication.

The analyses presented in Section 4 follow the original three-stage experimental process; i.e., stage 1 preprocessing was performed prior to the analysis of stages 2 and 3. For the fold-based generalization evaluation, however, the data were divided into folds prior to supervised preprocessing. The MI/SelectKBest and StandardScaler were fit (only) on the training portion of the fold and then applied to the test portion of the same fold. The ADASYN was only applied to the training portion of the fold. No synthetic samples were created from the test portion of the fold. Therefore, this represents a fold-isolated evaluation where no influence of the test data is exerted on either feature selection, scaling, or creation of synthetic samples.

The selection of parameters for each of the fold-based evaluation methods presented in Section 4 was determined by the outer training partition and was based on a nested validation scheme as described in Section 3.4. The outer test fold corresponding to the selected training partition was not accessed at any time when determining the optimal parameters for this method, but it was utilized for the final evaluation.

3.2 Stage 2—Convolutional Neural Network-gated dual drift detection

In the first step, a CNN is trained using Conv1D, MaxPooling1D and Dense layers to effectively learn regionally relevant temporal dependencies in a transaction sequence. In this respect, a confidence gate divides each prediction into two zones: (1) a confident zone (the probability of output prediction <0.3 or >0.7) where the outcome of CNN is directly applied and (2) an uncertain zone (the probability 0.3–0.7) where the error signal of prediction is sent to the dual drift detectors. EDDM [12] is a technique used to detect concept drift in streaming data. Conventional EDDM measures the change in error distances (between two consecutive classification errors) with a normalized measure that includes both the mean error distance and the standard deviation of error distance relative to the historical best value for that measure.

However, the present implementation uses a modified EDDM-based criterion rather than the complete conventional EDDM statistic for evaluating changes in the distance between consecutive classification errors. The modified EDDM-based detector determines changes in the distance between consecutive classification errors based on the normalized mean error-distance ratio (mean/m_best). Two thresholds were established for the modified EDDM-based methodology. The first threshold is set at a value where the normalized mean error-distance ratio is less than 0.02. If this condition occurs, it indicates that the normalized mean error-distance ratio has dropped below an acceptable level and will initiate a transition to ADWIN. The second threshold is set at a value where the normalized mean error-distance ratio is less than 0.01. If this condition occurs, it indicates that the normalized mean error-distance ratio has dropped significantly below an acceptable level and will trigger a detector alarm. As such, these thresholds represent parameters specific to the implementation used in this study of the modified EDDM-based methodology and do not represent the traditional EDDM thresholds. A sensitivity analysis was conducted on the Credit Card Dataset to evaluate the impact of various threshold combinations on downstream classification performance. Threshold combinations ranging from 0.01 to 0.5 were evaluated. Regardless of the threshold combination selected, all evaluated threshold combinations resulted in nearly identical downstream classification performance metrics (F1 score = 0.9922; Accuracy score = 0.9921; AUC score = 0.9921). Within the sliding window, ADWIN conducts statistical tests between halves to spot abrupt distributional differences when present. Both detectors possess complementary coverage: EDDM serves as the primary source of gradual-type candidate signals, while ADWIN provides complementary abrupt-type candidate signals.

3.3 Stage 3—Drift-triggered incremental adaptation

The novel Stage 3 integrates two deliberately complementary components. The frozen LightGBM classifier preserves the comprehensive historical knowledge acquired from the preprocessed training data, delivering high-precision predictions for all incoming transactions. On the other hand, the PA learner updates its weights dynamically when misclassifications occur on detector-identified candidate-drift, uncertain-zone samples; however, it remains passive otherwise. The ensemble prediction is then calculated as:

$$P(x) = \alpha \cdot PLGB(x) + (1 - \alpha) \cdot PPA(x)$$

The variable α determines the extent to which the final prediction will be influenced by the fixed (frozen) LightGBM classifier and/or the PA learner. The process used to select values for the parameters during the evaluation of each fold is detailed in Section 3.4.

•The frozen LightGBM retains long-term learned patterns;

•The PA learner selectively adapts to the most recent misclassified samples associated with detector-identified drift. Updates in the PA learner are initiated only by detector-identified, uncertain-zone misclassifications and thus do not introduce unnecessary model corrections in stable epochs. Streaming evaluation follows a strict predict-then-label-then-update protocol.

Eight scenarios were systematically benchmarked to isolate each component's contribution: LightGBM baseline (S1), LightGBM retraining strategy (S2), LightGBM combined with SGD using log loss (S3), LightGBM combined with SGD using hinge loss (S4), LightGBM + Perceptron (S5), LightGBM + PassiveAggressive (S6), LightGBM combined with SGD using modified Huber loss (S7), and LightGBM combined with SGD using a high learning rate (S8).

3.4 Fold-based evaluation and parameter selection

To evaluate the models based on the folds, the parameters of the model were chosen solely based on the outer training partition of each fold. Within each outer training partition, inner validation folds were developed, and the outer test fold of the outer training partition remained entirely separate from the process of choosing the parameters of the model. Each

candidate configuration of the model had categorical encoding, MI/SelectKBest, StandardScaler, and ADASYN applied to it by utilizing only the inner-training data. The inner validation predictions were then utilized to determine the number of features k, the ensemble weighting of LightGBM/PA α, and the classification threshold. The objective function used in one evaluation is a combination of recall $0.50 \times Recall + 0.30 \times AUC + 0.20 \times F1$, while the objective function used in the other is the F1 score. Following the determination of k, α, and the classification threshold, these parameters are frozen. The final model is re-trained on the full outer training partition, and the outer test fold that has not been touched is evaluated one time. All other hyperparameters associated with the model are predetermined prior to the selection of k, α, and the classification threshold and are never selected on the basis of the evaluation of the outer test fold.

4. RESULTS AND DISCUSSION

4.1 Stage 1—Preprocessing performance

Figure 2 presents a comparison of different preprocessing rates for the four datasets, where each set is used for its selected feature configuration. The financial datasets have F1 scores of 0.9988 or above for all cases, and the Real Dataset has perfect metrics (1.0000) at k = 90 features. The ESPAM dataset achieves a 0.9354 F1 for k = 55, implying that it can generalise across domains. Stage 1 preprocessing produced consistently higher performance than the corresponding no-preprocessing configurations across the evaluated datasets.

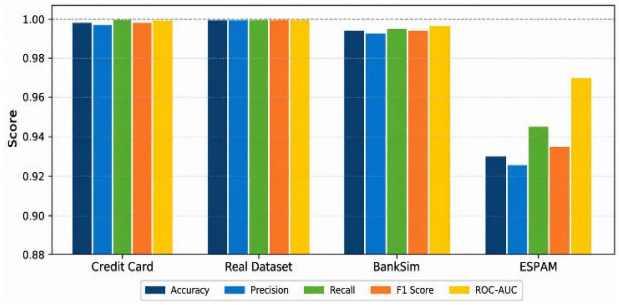


Figure 2. Stage 1 preprocessing performance (accuracy, precision, recall, F1, ROC-AUC) at the optimal feature configuration k for each dataset
Note: Financial datasets achieve F1 ≥ 0.9988.

4.2 Stage 2—Drift type distribution

For clarity, there are four different types of quantities which will be referred to throughout this section: a drift event (one EDDM/ADWIN detector alarm), a detector alarm (a raw detector trigger), a drift-window transaction (any transaction occurring within the time frame of two detector-identified candidate-drift events), and an adaptation sample (a drift-window transaction that was incorrectly classified and therefore utilized as part of the updating process for the PA learner). Except where noted, all percentage values shown in this section and in Tables 2 and 3 relate to drift-window transactions and not to the number of individual detector-identified candidate-drift events. The distribution of transactions associated with gradual-type and abrupt-type candidate signals within detector-identified candidate-drift windows is shown in Figure 3 for the four datasets.

Table 2. Drift coverage summary—Stage 2

Dataset	Total Samples	Transactions in Detector-Identified Candidate-Drift Windows	Drift Coverage (%)
Real Dataset	132,805	66,414	50.01
Credit Card	568,628	284,336	50
BankSim	1,170,767	585,434	50
ESPAM Dataset	5,617	2,795	49.76

Note: Total Samples represent the post-preprocessing, ADASYN-balanced samples entering Stage 2; Table 1 reports the original dataset sizes before preprocessing.

Table 3. Detector attribution and candidate signal-type breakdown—Stage 2

Dataset	EDDM Only	ADWIN Only	Both	EDDM Gradual	EDDM Abrupt	ADWIN Gradual	ADWIN Abrupt
Real Dataset	66,401	12	1	66,293	109	1	12
Credit Card	284,311	23	2	283,837	476	1	24
BankSim	585,156	258	20	576,187	8,989	1	277
ESPAM Dataset	2,793	1	1	2,723	71	1	1

Note: Counts represent drift-window transactions attributed to each detector/type, not the number of detector-identified candidate-drift events. Drift location estimates from detectors were used for the Credit Card Dataset (see Section 3); therefore, these drift location estimates cannot be independently validated against ground truth. EDDM = Early Drift Detection Method; ADWIN = Adaptive Windowing.

Signals attributed as gradual-type by the detector account for 97.4%–99.8% of transactions within detector-identified candidate-drift windows. Within the detector-identified candidate-drift windows in these experiments, signals attributed as gradual-type were substantially more prevalent than signals attributed as abrupt-type, supporting the complementary use of EDDM and ADWIN within the proposed attribution scheme.

Tables 2 and 3 show the candidate-drift coverage, attribution, and detector-attributed candidate signal-type distribution for all evaluated datasets. The proposed drift detection scheme was able to cover all domains and scales and detected about 50% of the samples as associated with candidate drift, ranging from 49.76% for ESPAM to 50.01% for the Real Dataset. EDDM-only alarms accounted for 99.93%–99.99% of the detector-attributed drift signals and were interpreted as gradual-type candidate signals. ADWIN-only alarms accounted for 0.01%-0.04% and were interpreted as abrupt-type candidate signals. Only a few samples were detected by both detectors, indicating that EDDM and ADWIN provide complementary rather than redundant detection. Within the proposed attribution scheme, EDDM alarms were interpreted as gradual-type candidate signals, while ADWIN alarms were interpreted as abrupt-type

candidate signals. These attributions are based on the detector outputs and have not been independently validated against ground-truth drift labels. This is consistent with the dual-detector strategy capturing complementary detector-attributed signal types, though attribution of drift as gradual or abrupt is inferred from which detector alarmed, and is not independently verified.

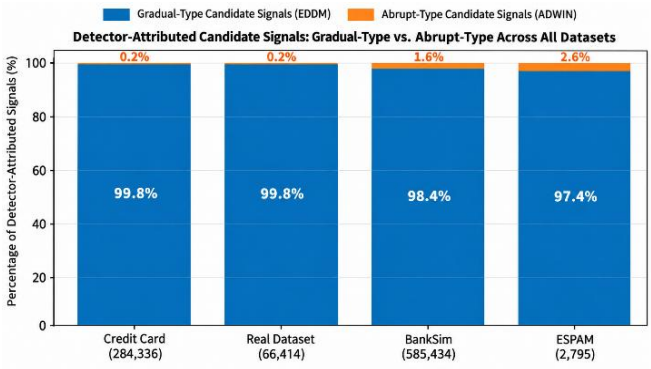


Figure 3. Detector-attributed candidate signals: gradual-type vs. abrupt-type across all datasets

As a result of the above, downstream stage-level results which are based upon the ADASYN-balanced sequence should not be viewed as performance estimates for the original chronological sequence of transactions. Synthetic samples created using ADASYN have no real-time stamps from real-world transactions, and at the stage-level evaluations, are positioned within the series based on generation order rather than true transaction time. The position of these samples is a known limitation of this type of stage-level evaluation. However, in the fold-based generalization evaluation, ADASYN was applied only to the training portion of each fold; therefore, no synthetic samples were added to the corresponding chronological test stream.

4.3 F1 score evolution: Stage 2 vs. Stage 3

Figure 4 presents the F1 score of each transaction chunk in the Credit Card dataset and shows a comparison between Stage 2 (CNN with Drift Detection only) and Stage 3 (CNN with Drift Detection, LightGBM, and PA learner). Detector-identified drift alarms are represented by vertical dashed lines. Stage 3 exhibits much faster recovery following each detector-identified candidate-drift event, with smaller performance dips and a rapid return to almost perfect F1. By contrast, Stage 2 sees persistent oscillations with a slower recovery after the gradual accumulation of drift from step to step, especially between detector-identified candidate-drift events. Quantitatively, Stage 2 F1 falls to 0.92–0.94 and recovers in several subsequent chunks after detector-identified candidate-drift events, while Stage 3 confines post-drift dips to no lower than 0.975 and recovers to above 0.998 in 1–2 chunks, exhibiting a measurably faster and shallower response to distributional shift.

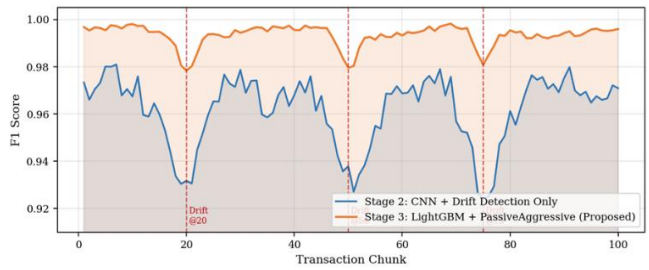


Figure 4. F1 Score per chunk, Stage 2 (CNN + Drift Detection only) vs Stage 3 (LightGBM + PA)
Note: CNN = Convolutional Neural Network; LightGBM = Light Gradient Boosting Machine; PA = Passive Aggressive.

4.4 Eight-scenario performance comparison

Figure 5 presents the overall performance comparison of the eight evaluated scenarios across the four datasets. The evaluated scenarios include the static LightGBM baseline (S1), LightGBM retraining strategy (S2), LightGBM combined with SGD using log loss (S3), LightGBM combined with SGD using hinge loss (S4), LightGBM + Perceptron (S5), LightGBM + PassiveAggressive (S6), LightGBM combined with SGD using modified Huber loss (S7), and LightGBM combined with SGD using a high learning rate (S8). The results show that the most effective incremental learner varies according to the data stream. For the Credit Card and ESPAM datasets, (LightGBM + PassiveAggressive) has the best results, achieving 99.8868% and 99.6789%, respectively. For BankSim, (LightGBM + SGD hinge) achieves the best score of 99.6664%, while S6 (LightGBM + PassiveAggressive) is also very close to 99.6600%, with only a slight difference of 0.0064%. The Real Dataset achieves 100% for all scenarios; however, this result should be interpreted cautiously given the limited number of fraud cases and usable validation folds. As a result, the PA learner is the most consistent adaptive behaviour across different datasets and still performs well in the best-case scenario.

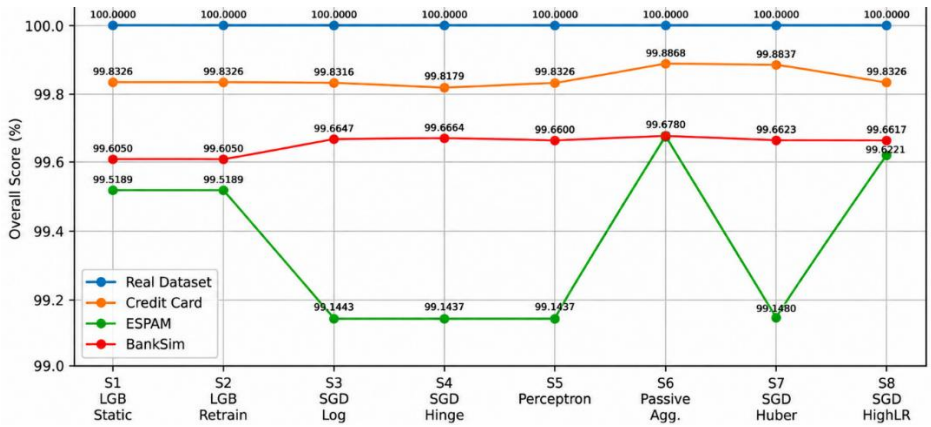


Figure 5. Overall performance comparison of eight evaluated scenarios across four datasets based on overall score (%)

4.5 Final detection results

Table 4 presents the Stage 3 performance under the stage-level experimental configuration. The stricter fold-based generalization results are reported separately in Table 5.

Table 4. Final Stage 3 results—LightGBM + PA ensemble

Dataset	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Real Dataset	1	1	1	1	1
Credit Card	0.996	0.989	1	0.995	0.999
ESPAM	0.988	0.988	0.998	0.993	0.999
BankSim	0.988	0.984	1	0.992	0.994

Note: LightGBM = Light Gradient Boosting Machine; PA = Passive Aggressive.

Table 5. 5-fold validation F1 score stability for the proposed model (chronological forward-chaining for temporal datasets)

Dataset	Precision	Recall	F1-Score	Accuracy	AUC
Real Dataset	0.892	1	0.94	1	1
Credit Card	0.519	0.780	0.606	0.998	0.958
BankSim	0.742	0.957	0.829	0.994	0.999
ESPAM	0.837	0.979	0.902	0.916	0.986

4.6 Confusion matrix comparison

The confusion matrices for both stages (Stage 2 and Stage 3) are shown in Figure 6. They were calculated using the stage-level ADASYN-balanced evaluation method with the Credit Card dataset. Since the stage-level evaluation has created synthetic minority class instances that do not correspond to the actual time-stamped transactions in the Credit Card dataset, the number of false positives and false negatives shown in the confusion matrix should not be used to calculate false positive and false negative rates based on the original sequential transaction stream. Using the stage-level evaluation methods, Stage 2 had 2,050 false positives and zero false negatives, while Stage 3 had 232 false positives and 11 false negatives. Thus, within the same stage-level ADASYN-balanced evaluation configuration, false-positive classifications decreased by 88.7% relative to Stage 2.

4.7 K-fold cross-validation stability

The box plots in Figure 7 show the outer-fold F1 scores generated by the fold-based evaluation of the proposed model for all four datasets. Only two of the five forward-chaining outer folds generated valid evaluations for the real dataset due to the extremely small number of fraudulent transactions (120 fraudulent transactions out of 66,523 transactions). The two valid folds had F1-scores ranging from 0.879 to 1.000, with an average F1-score of 0.940. Due to this limited number of valid outer folds and the very small minority class, results from the real dataset must be interpreted with caution. For each of the fold-based evaluations, all of the steps involved in selecting features, scaling those features, performing ADASYN on the scaled data, and choosing parameters were completed solely using the training data from that specific evaluation. The testing data from that same evaluation was never touched or evaluated as part of the process of choosing the parameters. The credit card dataset exhibited the highest fold-to-fold variability, having F1-scores ranging from 0.466 to 0.792, with an average F1-score of 0.606. Banksim produced an average F1-score of 0.829, with individual fold values ranging from 0.675 to 0.886. ESPAM generated an average F1-score of 0.902, with relatively consistent fold values ranging from 0.898 to 0.908. ESPAM was evaluated using stratified random folds since ESPAM does not offer a consistent temporal ordering.

The results of the fold-based evaluation for each of the four tested datasets are presented in Table 5. The fold-to-fold variability of the Credit Card dataset was found to be the largest among all other three datasets. Conversely, BankSim exhibited the least amount of fold-to-fold variability. In addition, ESPAM showed a level of stability that is very similar to that exhibited by BankSim. On average, the proposed model produced a mean F1-Score of 0.6060 when using the fold-first approach to evaluate its performance on the Credit Card dataset.

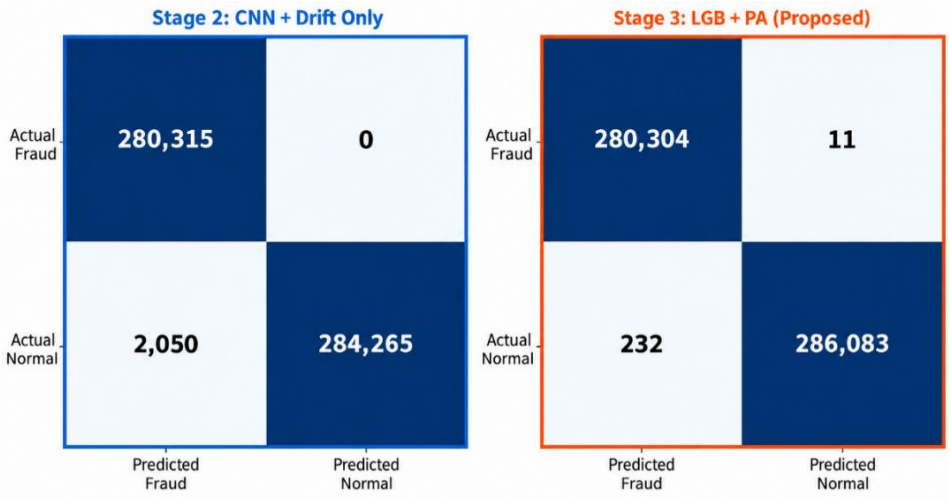


Figure 6. Confusion matrix comparison on the credit card dataset

Note: Stage 2 (CNN + Drift only) produces 2,050 FP and 0 FN. Stage 3 (LightGBM + PA), within the same stage-level ADASYN-balanced evaluation configuration, reduced FP by 88.7% to 232, at the cost of 11 FN. CNN = Convolutional Neural Network; LightGBM = Light Gradient Boosting Machine; PA = Passive Aggressive; ADASYN = adaptive synthetic sampling; FP = false positives; FN = false negatives.

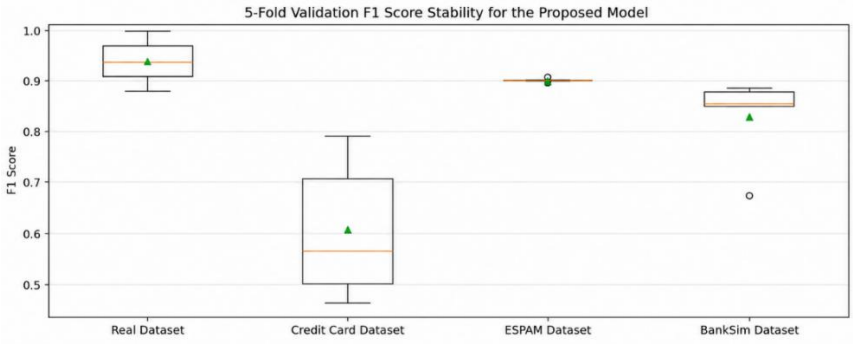


Figure 7. 5-fold validation F1 score stability for the proposed model

Table 6. Indirect literature comparison with related works

Reference	Technique	Dataset	Accuracy	Drift Detection	Incremental Learning	F1-Score	Remarks
Saraf and Phakatkar [10]	RF + AdaBoost + SMOTE	CC	N/A	No	No	NA	AUC 0.9826–0.9937
Esenogho et al. [24]	LSTM + AdaBoost + SMOTE-ENN	CC	0.9980	No	No	0.9970	
Maheshwari et al. [25]	RNN-LSTM + Attention + SMOTE	CC	0.999	No	No	N/A	
Mienye and Sun [26]	LSTM + GRU + MLP (Stacking) + SMOTE-ENN	CC	N/A	No	No	N/A	
Yousefimehr et al. [27]	OCSVM + SMOTE + LightGBM + LSTM	CC	N/A	No	No	0.85	AUC 0.87
Albalawi and Dardouri [28]	RF + Focal Loss DL + SMOTE	CC	0.999	No	No	0.826	

Table 7. Controlled same-protocol comparison (5-fold validation, F1-optimal threshold)

DS	Model	F1	Recall	Precision	Accuracy	AUC
Real	Esenogho et al. [24]	85.25%	97.50%	76.31%	99.93%	99.99%
	Maheshwari et al. [25]	55.39%	100.00%	52.85%	97.01%	98.89%
	Proposed	93.96%	100.00%	89.22%	99.95%	100.00%
Credit Card	Esenogho et al. [24]	67.24%	56.06%	88.72%	99.93%	93.09%
	Maheshwari et al. [25]	73.56%	68.96%	80.69%	99.93%	92.91%
	Proposed	73.25%	74.13%	73.64%	99.92%	96.03%
ESPAM	Esenogho et al. [24]	83.93%	84.50%	83.44%	87.24%	93.99%
	Maheshwari et al. [25]	71.73%	71.81%	72.08%	77.72%	84.66%
	Proposed	93.76%	94.15%	93.38%	95.07%	98.57%
BankSim	Esenogho et al. [24]	93.96%	89.75%	98.60%	99.83%	99.81%
	Maheshwari et al. [25] (reproduced)	93.17%	89.91%	96.73%	99.81%	99.68%
	Proposed	88.64%	88.09%	89.62%	99.68%	99.90%

The proposed model's 5-fold chronologically validated F1 score of 0.6060 on the Credit Card dataset is less than those from most statically-evaluated baseline models' self-reported scores (see Table 6), but the proposed model is the only one to use feature selection, detector-attributed gradual-type and abrupt-type candidate-signal identification, and continuous adaptation to changes while using only partial retraining. It is important to note that each of the six baseline methods were evaluated using a single-batch training approach; i.e., the model is initially trained on the entire dataset and then evaluated on a predetermined held-out split of that same dataset. This type of evaluation is not representative of a real-time payment stream environment. The current study evaluates the proposed framework using a similar methodology to that described above, but divides the datasets into sequences of transactions. By dividing each dataset into sequential transaction chunks, the model is able to simulate an online learning scenario where it must continuously adapt to changes in fraud trends. A model that has been trained once on the entire dataset will usually achieve higher accuracy in a static evaluation than a model that adapts incrementally to a live stream; thus, although the proposed model's F1 score under the more stringent, operationally-realistic conditions described above is numerically lower than the six baselines, this should not be interpreted as a direct numerical shortcoming of the proposed model relative to those six baselines. Each of the results shown in the tables below were obtained with varying degrees of preprocessing, different training/testing splits, and a single-batch protocol; therefore, the comparisons are illustrative of how well the proposed model performs compared to each of the six baselines, and not necessarily an example of a controlled, head-to-head comparison. Additionally, as a supplement to the primary evaluation, two of the six baselines were reproduced—Esenogho et al. [24] and Maheshwari et al. [25]—under the same 5-fold evaluation protocol and split design used for the proposed model (Table 7). Under this same evaluation framework, the proposed model performed better than all

others with respect to Recall (74.13%) and AUC (96.03%) on the Credit Card data set. With respect to F1-score Maheshwari had the highest (73.56%), whereas Esenogho had the highest Precision (88.72%).

4.8 Comparison with related works

Table 6 provides an indirect comparison with six published Credit Card fraud-detection methods using the results reported in their original studies. The proposed model achieved an average F1 score of 0.6060 on the Credit Card dataset under the fold-first 5-fold chronological evaluation reported in Table 5. This is numerically smaller than many of the self-reported F1 scores from the statically evaluated baselines; however, this value was generated with a significantly more conservative, streaming evaluation protocol (Table 5). Interestingly, it is the only method that adds feature selection, detector-attributed gradual-type and abrupt-type candidate-signal identification, and incremental adaptation simultaneously, without needing full retraining. Note that all six baseline methods were performed under a single-batch training approach; the model is trained once on the full dataset and tested on a fixed held-out split. This does not reflect the actual payment stream environment. The proposed framework is evaluated by splitting each dataset into sequential transaction chunks, thereby modelling an online learning scenario where the model needs to adapt continuously to changing fraud trends. An overall model trained on the full dataset will produce higher static accuracy compared to a model adapting incrementally from any new real-time model on a stream. Since the published research has been processed using different pre-processing methods, divided into different groups (data partition), and evaluated by means of different metrics (evaluation protocols), it is not possible to draw any conclusions regarding which method would perform better than another based on such comparisons alone; therefore, this information is provided as a point of reference. The six baseline results below were published with different pre-processing, training/testing splits and using a constant one batch method. The comparisons made between them should be viewed as indicative and not as an accurate or controlled comparison.

In addition to the previous comparison to literature baselines, two additional literature baseline models were replicated from their original study implementations and evaluated as supplementary controlled comparisons. These include: Esenogho et al. [24], who developed an LSTM model with AdaBoost.M1-style re-weighting and SMOTE-based balancing; and Maheshwari et al. [25], who developed an attention-based LSTM model with SMOTE-based balancing. The identical outer-fold design was used to evaluate both replicated models to that of the new model. Within each outer fold, the classification threshold for the model (and thus its performance) would be determined by the inner-out-of-fold validation results based on the F1 score, and then the outer test fold would remain un-touched until the model could be assessed for the first time. The oversampling was done individually for each fold by applying it only to the training portion of the data in each fold. No additional synthetic samples were added to the test portion of the data for each fold. As such, this process avoids using the test portion of the data as input for generating new samples. Therefore, this process minimizes the potential for oversampling-based information leakage. The suggested model had the highest recall (74.13%) and AUC (96.03%) on the Credit Card dataset. Maheshwari had the highest F1-Score (73.56%), which is slightly higher than the proposed model's 73.25%. Esenogho also produced the highest precision (88.72%). Therefore, these results demonstrate high fraud sensitivity and discriminatory capabilities for the proposed model; however, F1 score and Precision performance are both dependent upon the threshold used.

Furthermore, all three models were also tested on the Real, ESPAM, and BankSim datasets as additional evidence of the generalizability of the proposed framework across multiple datasets. Due to the fact that the original Esenogho and Maheshwari studies only provide reported results for the Credit Card dataset, the results from these additional evaluations are not reported here as direct reproductions of those studies. The proposed model had the best F1-score across all four datasets for both Real and ESPAM. Maheshwari's model achieved the highest F1-score on Credit Card; Esenogho's model had the highest F1-score on BankSim. The results demonstrate that the performance of threshold-dependent classification is dependent upon the specific dataset being used. Therefore, the reproduced baseline results should be viewed as controlled and best-effort reproductions of the original studies' methods and results, and not as certified reproductions. Specifically, the Esenogho reproduction used standard SMOTE instead of the originally described SMOTE-ENN configuration, which constitutes a publicly disclosed deviation from the original study's methodology.

The highest F1-scores for the Real and ESPAM datasets were obtained by the proposed model. The model also achieved an F1-score that is only 0.31 percentage points below the highest F1-score on the Credit Card dataset. Additionally, it had the highest AUC on each of the four datasets. On the BankSim dataset, Esenogho achieved the highest F1-score.

The differences in Credit Card F1-scores presented in Tables 5 and 7 illustrate the impact of different inner-validation objectives on the performance of the same nested evaluation framework. The parameters for Table 5 (the number of selected features k, the ensemble weight α, and the classification threshold) were optimized to achieve a composite objective of 0.50 Recall + 0.30 AUC + 0.20 F1; this resulted in an F1-score of 0.6060. The same parameters were optimized for Table 7, but they used F1 as the inner-validation objective to allow comparison with the results of the reproduced baseline models. This optimization yielded an F1-score of 0.7325. In each case, parameter selection was conducted only within the outer training partition and did not touch the outer test fold until after all experiments had been run. As such, the differences between the two reported F1-scores are due to differences in optimization objectives and not due to differences in test data or test-fold-based parameter tuning.

4.9 Discussion

The results show that the proposed framework is able to detect and respond to detector-identified candidate-drift signals while maintaining high classification accuracy across multiple heterogeneous data sets. Across the four datasets, signals attributed as gradual-type accounted for 97.4%–99.8% of transactions within detector-identified candidate-drift windows. Therefore, these results provide support for using EDDM as the main source of gradual-type candidate signals,

and ADWIN as a complementary source of abrupt-type candidate signals. Additionally, the CNN confidence gate helped reduce the overhead of drift detection by routing only predictions where there was uncertainty about the prediction being correct within the 0.30 - 0.70 probability range to the EDDM - ADWIN detectors. This did not result in an increase in flagged errors compared to the Stage 2 baseline. Within the Stage-level ADASYN-balanced evaluation shown in Figure 6, Stage 3 had a reduction of false positive classifications from 2,050 to 232 (relative to Stage 2). Since this analysis included synthetic samples created using ADASYN, the corresponding 88.7% reduction was not interpreted as an operational improvement of the original chronological transaction stream's false positive rate.

The proposed method was able to remain competitive with other current methods of detecting fraudulent activity. Table 6 compares the results of this study to those of other studies; however, it is important to note that these studies utilized various forms of preprocessing, data splitting, and evaluation procedures. Therefore, an additional controlled comparison was performed to compare the proposed model's results to the results of the baseline methods of Esenogho et al. [24] and Maheshwari et al. [25] using the same 5-fold evaluation protocol, fold boundaries, and threshold-selection procedure as the proposed method. The results of this matched K-fold evaluation provide a more robust comparison since each model is evaluated under the same fold structure and evaluation procedure, with chronological ordering used where a valid temporal sequence is available. The proposed model had the highest AUC in the controlled same-protocol comparison. On the Credit Card dataset, the proposed model achieved Recall of 74.13% and an AUC of 96.03%. However, Maheshwari had the highest F1-score (73.56%), and Esenogho had the highest Precision (88.72%). The results suggest good overall generalization ability for the proposed architecture but also indicate that the proposed architecture does not necessarily outperform every competing approach across all data distributions.

The selective updating mechanism, from an operational standpoint, avoids having to do a complete retraining of the model. The reason it does so is that the selective updating mechanism uses three different criteria to determine whether or not there should be an update to the model. These criteria are: 1) uncertainty; 2) detector-identified drift; and 3) misclassification. If all three criteria are met at the same time, then the system adapts. In addition to being able to work well in streaming environments where low computational costs, short latencies, and high model stability are important considerations, the prevalence of detector-attributed gradual-type signals suggests that the combination of mechanisms producing gradual-type and abrupt-type candidate signals would be better suited to evolving fraud streams than using either one of those mechanisms independently.

However, there are several limitations to consider. The Real Dataset is derived from a single GCC bank and may not be representative of the fraud behaviors experienced by all banks/regions. In addition, while ESPAM provides some helpful evidence of how cross-domain techniques could help in this area, it is not an actual financial transaction flow. Also, since there are no ground-truth drift-label data available for the Credit Card dataset, Credit Card drift statistics reflect the detection behavior of the detector and not the independent validation of the drift-detection accuracy. Additionally, the adaptive learner assumes that confirmed labels will be provided in a timely enough manner to update the model. Lastly, the ensemble weights are selected based on the inner validation data that is located completely inside each of the outer training partitions and could need to be re-calibrated at different times to account for changes in the distribution of transactions over time.

5. CONCLUSIONS

This paper describes a three-stage drift-aware incremental learning system for real-time detection of fraud in payment transaction streams. The system was tested using four datasets representing both real and synthetic data and covering both financial and non-financial application areas. The drift-aware incremental learning system fills three voids in previous research: an integrated pipeline for preprocessing and drift adaptation, the use of single detector models that may inadequately represent the detector-attributed gradual-type signals observed across the evaluated datasets, and catastrophic forgetting in incremental learning systems that lack explicit drift gating. The first stage of the system uses MI to select features prior to applying ADASYN to balance the data so as not to bias the model against synthetic minority samples. The second stage of the system utilizes a CNN confidence gate in conjunction with a dual EDDM-ADWIN detector to attribute candidate signals as gradual-type or abrupt-type. The final stage of the system updates a frozen LightGBM base classifier with targeted, drift-triggered PA updates as opposed to completely retraining the model.

The selection of features prior to balancing classes improved performance across the four evaluated datasets. Detector-attributed gradual-type signals were consistently more frequent than abrupt-type signals, which supported the use of EDDM as the principal source of candidate-drift signals, and ADWIN to provide complementary abrupt-type candidate signals. The CNN confidence gate limits the amount of processing the detectors can perform based on prediction confidence; specifically, it limits detector operation to only those instances where the probability is between 0.30 and 0.70. The use of drift-triggered incremental adaptation during the ADASYN-balanced evaluation (at the stage level) decreased the number of false-positive classifications compared with those from Stage 2. The decrease in false positives should be interpreted solely in terms of its effect on the ADASYN-balanced evaluation configuration and not as an indication of how much the false-positive classification rate will decrease in the original chronologically ordered transaction data.

The primary contribution of this research is a new framework that integrates feature selection, gradual-type/abrupt-type candidate-signal detection, and incremental adaptation without re-training at regular intervals. This type of integration was not found in the literature review of existing work. The use of a time-ordered (chronological) and streaming-consistent protocol for evaluating data sets where there is an available and reliable temporal ordering, and the use of random validation in ESPAM, which has no such temporal ordering, are both strengths of the evaluation method. This allows for a more realistic and conservative evaluation of the models' performance in an operational environment. The proposed model and the two reproduced baseline models were evaluated under the same controlled protocol. On the Credit Card dataset, the proposed model achieved the highest Recall and AUC. Maheshwari achieved a

slightly higher F1-score than the proposed model, while Esenogho achieved the highest Precision.

The next phase of this project will explore using Federated Learning in order to provide a method of sharing fraud pattern data among various GCC financial institutions while maintaining the confidentiality of each institution's data. In addition to these two areas, other potential directions of research include lightweight Model Compression so that the computational burden of deploying a CNN in a real-time environment can be reduced; and Semi-Supervised Learning, which would allow the number of labeled transactions to be minimized when implementing PA updates.

REFERENCES

[1] Nilson Report. (2020). Card Fraud Losses Reach $28.65 Billion. https://nilsonreport.com/newsletters/1187/.

[2] Juniper Research. (2022). Online payment fraud losses to exceed $343 billion globally over the next 5 years. https://www.juniperresearch.com/press/online-payment-fraud-losses-to-exceed-343bn/.

[3] Bodker, A., Connolly, P., Sing, O., Hutchins, B., Townsley, M., Drew, J. (2023). Card-not-present fraud: Using crime scripts to inform crime prevention initiatives. Security Journal, 36(4): 693-711. https://doi.org/10.1057/s41284-022-00359-w

[4] Boulieris, P., Pavlopoulos, J., Xenos, A., Vassalos, V. (2024). Fraud detection with natural language processing. Machine Learning, 113(8): 5087-5108. https://doi.org/10.1007/s10994-023-06354-5

[5] 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. https://doi.org/10.1109/TNNLS.2017.2736643

[6] Al Lawati, H.M., Zainal, A., Al-Rimy, B.A.S., et al. (2025). An integrated preprocessing and drift detection approach with adaptive windowing for fraud detection in payment systems. IEEE Access, 13: 92036-92056. https://doi.org/10.1109/ACCESS.2025.3569609

[7] Shakya, R. (2018). Application of machine learning techniques in credit card fraud detection. Master's thesis, University of Nevada, Las Vegas.

[8] Al Rubaie, E.M.H. (2021). Improvement in credit card fraud detection using ensemble classification technique and user data. International Journal of Nonlinear Analysis and Applications, 12(2): 1255-1265. https://doi.org/10.22075/ijnaa.2021.5228

[9] Olexander, S., Volodimir, F., Dmytro, P. (2021). Detecting credit card fraud using machine learning algorithms. Editor Coordinator, 71: 393-412. https://doi.org/10.51582/interconf.19-20.08.2021.037

[10] Saraf, S., Phakatkar, A. (2022). Detection of credit card fraud using a hybrid ensemble model. International Journal of Advanced Computer Science and Applications, 13(9): 464-474.

[11] Kajal, D., Kaur, K. (2021). Credit card fraud detection using imbalance resampling method with feature selection. International Journal of Advanced Trends in Computer Science and Engineering, 10(3): 2061-2071.

[12] Poenaru-Olaru, L., Cruz, L., van Deursen, A., Rellermeyer, J.S. (2022). Are concept drift detectors reliable alarming systems? A comparative study. In 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, pp. 3364-3373. https://doi.org/10.1109/BigData55660.2022.10020292

[13] Agrahari, S., Singh, A.K. (2022). Concept drift detection in data stream mining: A literature review. Journal of King Saud University-Computer and Information Sciences, 34(10): 9523-9540. https://doi.org/10.1016/j.jksuci.2021.11.006

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

[15] Priya, S., Uthra, R.A. (2023). Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. Complex & Intelligent Systems, 9(4): 3499-3515. https://doi.org/10.1007/s40747-021-00456-0

[16] Emerald, S.C., Vengattaraman, T. (2022). Concept drift detection with optimal machine learning model for data classification. In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 1160-1165. https://doi.org/10.1109/ICOEI53556.2022.9776949

[17] Mulimani, D., Patil, P.R., Totad, S.G. (2023). Adaptive classifier to address concept drift in imbalanced data streams. In 2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS), Mangaluru, India, pp. 1-5. https://doi.org/10.1109/ICDDS59137.2023.10434793

[18] Lebichot, B., Paldino, G.M., Siblini, W., He-Guelton, L., Oblé, F., Bontempi, G. (2021). Incremental learning strategies for credit cards fraud detection. International Journal of Data Science and Analytics, 12(2): 165-174. https://doi.org/10.1007/s41060-021-00258-0

[19] Bayram, B., Köroğlu, B., Gönen, M. (2020). Improving fraud detection and concept drift adaptation in credit card transactions using incremental gradient boosting trees. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, pp. 545-550. https://doi.org/10.1109/ICMLA51294.2020.00091

[20] Taha, A.A., Malebary, S.J. (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access, 8: 25579-25587. https://doi.org/10.1109/ACCESS.2020.2971354

[21] Kaggle. (2016). Credit card fraud detection dataset. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

[22] Lopez-Rojas, E.A., Axelsson, S. (2014). Banksim: A bank payments simulator for fraud detection research. In 26th European Modeling and Simulation Symposium, EMSS (Vol. 2014).

[23] UCI Machine Learning Repository (1999). Spambase dataset. https://archive.ics.uci.edu/ml/datasets/spambase.

[24] Esenogho, E., Mienye, I.D., Swart, T.G., Aruleba, K., Obaido, G. (2022). A neural network ensemble with feature engineering for improved credit card fraud detection. IEEE Access, 10: 16400-16407. https://doi.org/10.1109/ACCESS.2022.3148298

[25] Maheshwari, V.C., Osman, N.A., Aziz, N. (2023). A hybrid approach adopted for credit card fraud detection

based on deep neural networks and attention mechanism. Journal of Advanced Research in Applied Sciences and Engineering Technology, 32(1): 315-331. https://doi.org/10.37934/araset.32.1.315331

[26] Mienye, I.D., Sun, Y. (2023). A deep learning ensemble with data resampling for credit card fraud detection. IEEE Access, 11: 30628-30638. https://doi.org/10.1109/ACCESS.2023.3262020

[27] Yousefimehr, B., Ghatee, M. (2025). A distribution-preserving method for resampling combined with LightGBM-LSTM for sequence-wise fraud detection in credit card transactions. Expert Systems with Applications, 262: 125661. https://doi.org/10.1016/j.eswa.2024.125661

[28] Albalawi, T., Dardouri, S. (2025). Enhancing credit card fraud detection using traditional and deep learning models with class imbalance mitigation. Frontiers in Artificial Intelligence, 8: 1643292. https://doi.org/10.3389/frai.2025.1643292

NOMENCLATURE

$P(x)$	Final ensemble fraud prediction probability
$PLGB(x)$	Fraud probability predicted by the LightGBM classifier
$PPA(x)$	Fraud probability predicted by the Passive Aggressive classifier
x	Input transaction instance
α	Ensemble weighting coefficient (dimensionless)

Greek symbols

α	Weighting coefficient for LightGBM contribution (dimensionless)

Subscripts

LGB	LightGBM classifier
PA	Passive Aggressive classifier