10.48047/jocaaa.2024.33.04.55

Continual Learning Systems for Adaptive Fraud-Pattern Detection in High-Velocity Transaction Streams

Gopinathan Rathinavelu
GTSS Inc, Data Architect, Arizona, USA
Email ID: gopinathan.r@gmail.com

Abstract

High-velocity financial transaction streams exhibit severe class imbalance, evolving fraud strategies, delayed verification labels, temporal dependencies, and frequent changes in legitimate customer behaviour. Conventional batch-trained fraud detection models can therefore experience progressive performance degradation when previously learned fraud patterns become obsolete. This paper proposes a continual learning framework for adaptive fraud-pattern detection that integrates streaming data processing, concept-drift detection, incremental model updating, temporal feature engineering, and adaptive ensemble learning. The framework continuously evaluates incoming transactions, identifies changes in transaction distributions, selectively updates model parameters, and retains relevant historical knowledge while limiting catastrophic forgetting. Performance is assessed through fraud detection accuracy, precision, recall, F1-score, geometric mean, false-positive rate, detection latency, and adaptation efficiency. The proposed framework provides a systematic architecture for maintaining fraud detection capability under non-stationary transaction environments and supports scalable near-real-time financial intelligence.

Keywords: *Continual Learning, Fraud Detection, Concept Drift, Transaction Streams, Online Learning, Adaptive Machine Learning*

1. Introduction

The rapid expansion of electronic payments, mobile banking, digital wallets, online commerce, and real-time payment infrastructures has transformed financial transactions into continuously generated data streams. Unlike conventional static datasets, transaction streams exhibit high velocity, temporal dependence, evolving customer behaviour, extreme class imbalance, and continuously changing fraudulent strategies. A fraud detection system operating in such an environment must therefore identify suspicious transactions rapidly while adapting to changes in the underlying data-generating process. Traditional machine-learning models trained once on historical transaction records can gradually lose effectiveness because the statistical characteristics of both legitimate and fraudulent transactions are not stationary. In credit-card fraud detection, changes in customer spending habits and fraudster behaviour constitute significant sources of concept drift, while genuine transactions generally outnumber fraudulent transactions by a very large margin [11].

The operational characteristics of real-world fraud detection make the problem substantially different from conventional binary classification. In practical systems, transaction labels are frequently unavailable at the time of prediction because suspected fraud may require investigation or confirmation from customers and financial institutions. Consequently, the learning algorithm must operate with incomplete, delayed, and highly imbalanced feedback. Dal Pozzolo et al. demonstrated that realistic fraud detection requires explicit consideration of concept drift, class imbalance, and verification latency rather than assuming that complete and immediately available labels exist for all transactions [11]. These characteristics create a requirement for learning mechanisms that can continuously incorporate new information without repeatedly rebuilding a model from the entire historical dataset.

10.48047/jocaaa.2024.33.04.55

Streaming-oriented fraud detection has consequently emerged as an important research direction. Carcillo et al. developed the SCARFF framework to process large-scale credit-card transaction streams using distributed data-processing technologies and machine-learning techniques capable of addressing non-stationarity, class imbalance, and delayed feedback [12]. Their framework demonstrated that fraud detection can be integrated with continuous data ingestion, online feature engineering, distributed storage, and incremental classification. The study also highlighted the importance of scalability because real-world transaction streams can contain millions of transactions and require near-real-time risk assessment [12].

Concept drift represents a central challenge in such environments. In a stationary transaction environment, the statistical characteristics of incoming observations remain relatively stable. In a dynamic financial environment, however, the underlying transaction distribution can change substantially over time. Such changes may originate from altered consumer behaviour, emerging fraud mechanisms, seasonal purchasing patterns, technological developments, changes in merchant behaviour, or modifications in financial-service processes. Consequently, a model optimized using historical transaction data may gradually become less suitable for detecting contemporary fraudulent activity.

Conventional retraining strategies provide only a partial solution because frequent complete retraining can impose substantial computational and operational costs. More importantly, repeatedly replacing an existing model with a model trained predominantly on recent observations can result in the loss of historically relevant fraud knowledge. This creates a fundamental stability–plasticity problem: the detection system must possess sufficient adaptability to learn emerging fraud patterns while maintaining sufficient stability to preserve previously acquired knowledge. Continual learning provides a suitable conceptual framework for addressing this requirement by allowing a model to update its knowledge sequentially as new observations become available.

For high-velocity transaction streams, continual learning can be understood as a sequence of evolving learning environments. Rather than independently training a new classifier whenever transaction characteristics change, the existing model is progressively updated using new information. This enables the system to incorporate emerging fraud characteristics while retaining information acquired during earlier transaction periods. Such a mechanism is particularly relevant to financial fraud because historical fraud patterns may reappear after periods of inactivity, making complete replacement of previously acquired knowledge undesirable.

The requirement for adaptive learning becomes particularly important because fraud is inherently adversarial. Fraudsters can modify transaction amounts, merchant selection, geographical patterns, transaction timing, device characteristics, account behaviour, and transaction sequences to evade established detection mechanisms. At the same time, legitimate customers can change their behaviour because of travel, employment changes, seasonal spending, new payment methods, or changes in purchasing preferences. Therefore, a fraud detection model must distinguish genuine behavioural evolution from fraudulent behavioural changes rather than treating every distributional change as evidence of fraud.

Feature engineering is also critical in an adaptive transaction environment. Static transaction attributes such as amount, merchant category, transaction location, and payment method can be complemented by temporal and behavioural features derived from recent transaction history. Bahnsen et al. demonstrated the importance of feature-engineering strategies in credit-card fraud detection, particularly for representing transaction behaviour in ways that improve discrimination between legitimate and fraudulent activities [10]. In a continual-learning

10.48047/jocaaa.2024.33.04.55

framework, these features can themselves be updated over time, allowing the representation of customer behaviour to evolve with the transaction stream.

The proposed research therefore focuses on the development of a **continual learning system for adaptive fraud-pattern detection in high-velocity transaction streams**. The central objective is to construct a learning architecture capable of continuously processing transactions, detecting changes in fraud distributions, updating predictive models, retaining relevant historical knowledge, and maintaining low-latency fraud classification. The framework integrates streaming data processing, temporal feature engineering, concept-drift detection, incremental model adaptation, class-imbalance handling, and knowledge-retention mechanisms.

The research is motivated by the need to move beyond static fraud classifiers toward adaptive systems that treat transaction streams as continuously evolving environments. The principal research problem concerns how a fraud detection system can continuously learn emerging fraud patterns from high-velocity transaction streams while preserving previously acquired knowledge and maintaining computationally efficient real-time detection. Addressing this problem requires simultaneous consideration of predictive performance, adaptation speed, false-positive control, computational latency, class imbalance, delayed labels, and catastrophic forgetting.

Accordingly, the study develops an adaptive continual-learning architecture in which incoming transactions are processed sequentially, relevant behavioural characteristics are extracted, distributional changes are monitored, and model parameters are selectively updated when sufficient evidence of environmental change is identified. The resulting framework aims to establish a balance between rapid adaptation and historical knowledge retention. Such an approach can provide a foundation for fraud detection systems capable of operating under realistic non-stationary transaction conditions rather than relying exclusively on periodically retrained static models.

2. Literature Review

2.1 Machine Learning-Based Financial Fraud Detection

Machine learning has become an important component of financial fraud detection because of its ability to identify nonlinear relationships between transaction attributes and fraudulent behaviour. Supervised classifiers, ensemble learning, neural networks, anomaly detection, and cost-sensitive learning have been investigated for identifying suspicious transactions. However, the extreme imbalance between legitimate and fraudulent transactions makes conventional accuracy an inadequate performance measure. A classifier can obtain high overall accuracy while failing to identify a substantial proportion of fraudulent transactions. Dal Pozzolo et al. emphasized that realistic evaluation must account for the operational characteristics of fraud detection, including class imbalance, concept drift, and delayed verification information [11].

Feature engineering represents another important dimension of fraud detection. Transaction amount, merchant category, temporal information, geographical characteristics, and behavioural history can be transformed into discriminative features that improve fraud identification. Bahnsen et al. investigated feature-engineering strategies specifically for credit-card fraud detection and demonstrated the importance of constructing informative behavioural representations rather than relying exclusively on raw transaction attributes [10]. Such representations become even more important when the system operates continuously because behavioural characteristics can change over time.

10.48047/jocaaa.2024.33.04.55

Fraud detection is consequently not simply a conventional binary classification problem. The financial consequences of failing to detect fraudulent activity can differ substantially from those associated with incorrectly flagging legitimate transactions. Therefore, fraud detection systems must simultaneously consider fraud recall, precision, false-positive behaviour, detection latency, and operational cost. This requirement encourages the development of models that are optimized according to the practical characteristics of financial transaction environments rather than relying exclusively on general classification accuracy.

2.2 Streaming and Online Fraud Detection

The transition from batch transaction processing toward continuously generated transaction streams has created the need for streaming fraud detection architectures. Conventional machine-learning workflows generally assume that a complete training dataset is available before model development. In contrast, streaming systems must process observations sequentially while maintaining bounded computational and memory requirements.

Carcillo et al. addressed this problem through the SCARFF framework, which combines streaming data processing with machine learning for scalable credit-card fraud detection [12]. The framework incorporated distributed data ingestion, online feature engineering, storage, and classification, while explicitly addressing non-stationarity, class imbalance, and delayed feedback. Its evaluation on a large real-world transaction stream demonstrated the feasibility of scalable fraud detection under high-volume operating conditions [12].

Earlier research by Dal Pozzolo et al. also investigated credit-card fraud detection under delayed supervised information and concept drift [11]. Their research recognized that investigators cannot immediately verify every transaction and that the information eventually returned to the learning system may arrive after a substantial delay. Consequently, an online fraud detector must distinguish between the time at which a transaction is scored and the later time at which its actual class becomes known.

This characteristic fundamentally changes the learning process. A transaction may have already influenced financial decisions before its fraud status is confirmed. The model must therefore maintain a mechanism for incorporating delayed labels when they become available without disrupting the continuous transaction-processing pipeline. Streaming fraud detection consequently requires coordination between real-time scoring, historical transaction storage, label management, and incremental model updating.

2.3 Concept Drift in Transaction Streams

Concept drift occurs when the statistical characteristics of a data-generating process change over time. In fraud detection, this phenomenon may arise from changes in consumer behaviour, payment technologies, fraudster strategies, merchant characteristics, or institutional countermeasures. The resulting drift may be sudden, gradual, incremental, or recurring.

Dal Pozzolo et al. identified concept drift as one of the fundamental characteristics of real-world credit-card fraud detection [11]. Their analysis showed that both customer habits and fraudulent strategies evolve over time, making static assumptions about transaction distributions unrealistic. Carcillo et al. subsequently incorporated concept-drift handling into a scalable streaming architecture using adaptive mechanisms for continuously changing transaction environments [12].

Concept drift creates a direct challenge for model validity. A model trained during one period may accurately identify the relationships present during that period but gradually become less effective as the underlying transaction environment changes. The problem is particularly

10.48047/jocaaa.2024.33.04.55

significant in fraud detection because changes can be generated deliberately by adversarial actors attempting to avoid established detection rules.

Concept drift can also occur without any malicious activity. For example, seasonal shopping patterns, changes in consumer payment preferences, introduction of new payment channels, and changes in merchant behaviour can alter the characteristics of legitimate transactions. Therefore, an adaptive fraud detection system must differentiate between normal population changes and changes that correspond to emerging fraud patterns.

2.4 Incremental and Continual Learning

Incremental learning provides a mechanism through which a model can incorporate newly available observations without reconstructing the entire training process. This is particularly appropriate for high-velocity transaction streams because the volume of incoming data can make repeated full-scale retraining computationally expensive.

Continual learning extends this principle by emphasizing the preservation of knowledge acquired from previous learning stages while enabling adaptation to new distributions. For fraud detection, this produces a stability–plasticity trade-off. Excessive stability can prevent the system from learning newly emerging fraud patterns, whereas excessive adaptation can cause the model to rapidly overwrite previously useful fraud knowledge.

The knowledge-retention problem is particularly important in financial fraud detection because fraud patterns can be recurrent. A fraud strategy that disappears for several months may reappear after fraudsters modify their operational approach. Therefore, discarding historical information solely because it is temporally distant can reduce the ability of the model to recognize recurring fraud behaviour.

Continual learning is therefore particularly suitable for fraud detection when combined with selective memory mechanisms. Historical observations can be prioritized according to their relevance, recency, rarity, fraud characteristics, or contribution to model performance. Such an approach provides an alternative to both complete historical retraining and purely recent-data-based learning.

2.5 Class Imbalance and Delayed Feedback

Extreme class imbalance is one of the defining characteristics of transaction fraud detection. Legitimate transactions normally constitute the overwhelming majority of observations, whereas confirmed fraudulent transactions represent only a small fraction. This imbalance can cause conventional classifiers to prioritize the majority class and consequently overlook important fraudulent patterns.

Dal Pozzolo et al. investigated realistic learning strategies for highly imbalanced and non-stationary credit-card transaction streams and demonstrated the importance of considering class imbalance and concept drift simultaneously [11]. Their research showed that fraud detection models must be evaluated according to their ability to identify rare fraudulent events rather than simply their overall classification accuracy.

Delayed feedback represents a second major challenge. Fraud labels are often generated after customer complaints, transaction investigations, bank verification procedures, or other confirmation processes. Consequently, the learning system may not immediately know whether an incoming transaction was fraudulent.

This creates a temporal mismatch between prediction and learning. The model must generate a fraud-risk decision when the transaction arrives and subsequently incorporate the verified

10.48047/jocaaa.2024.33.04.55

outcome when it becomes available. A continual-learning architecture must therefore maintain appropriate historical records and associate delayed labels with the original transaction observations.

Class imbalance and delayed feedback also interact with concept drift. If the characteristics of fraudulent transactions change while the model is receiving only a limited number of confirmed fraud labels, the system may require more sophisticated mechanisms for detecting emerging patterns. This reinforces the need for adaptive sampling, cost-sensitive learning, drift detection, and selective memory management.

2.6 Research Gap and Proposed Research Direction

The existing literature establishes several important foundations for adaptive fraud detection. Machine-learning research has demonstrated the effectiveness of predictive and ensemble approaches, feature-engineering studies have improved transaction representation, and streaming frameworks have demonstrated scalable processing of large transaction volumes [10], [12]. Research on realistic fraud detection has further established the importance of concept drift, class imbalance, and delayed feedback [11].

Nevertheless, a methodological gap remains in integrating these requirements into a unified continual-learning framework. Many conventional fraud detection models remain predominantly batch-oriented, while streaming frameworks often emphasize data-processing scalability and drift adaptation rather than explicit long-term knowledge retention. Generic continual-learning approaches, meanwhile, are not inherently designed for the simultaneous challenges of extreme class imbalance, delayed verification, adversarial pattern evolution, and low-latency transaction scoring.

The research gap can therefore be characterized through five interconnected requirements: continuous adaptation, historical knowledge preservation, reliable concept-drift identification, computational efficiency, and robust handling of highly imbalanced fraud data. Addressing these requirements independently may not provide an effective solution because improvements in one dimension can negatively affect another. For example, aggressive adaptation may improve detection of new fraud patterns but increase forgetting of previously learned patterns, whereas excessive historical retention may reduce responsiveness to newly emerging fraud mechanisms.

The proposed study addresses this gap by developing a continual-learning architecture that continuously monitors transaction distributions, detects meaningful behavioural changes, updates the fraud classifier selectively, and preserves historically important fraud knowledge. Rather than replacing the complete model after every detected change, the proposed approach emphasizes controlled incremental adaptation.

The research consequently positions continual learning as an adaptive operational paradigm for fraud detection. The intended system combines streaming transaction processing, temporal feature engineering, drift-aware updating, class-sensitive learning, and knowledge-retention mechanisms to maintain detection capability across evolving transaction environments. This provides the methodological foundation for the subsequent development of the proposed framework, experimental methodology, performance evaluation, and comparative analysis.

10.48047/jocaaa.2024.33.04.55

3. Proposed Continual Learning Framework for Adaptive Fraud Detection

3.1 High-Velocity Transaction Stream Acquisition and Preprocessing

The proposed framework is designed for continuous processing of high-velocity financial transaction streams in which transactions arrive sequentially and require rapid classification. The architecture considers each transaction as a time-dependent observation containing transactional, behavioural, temporal, geographical, device-related, and merchant-level information. Real-world fraud detection systems must operate under non-stationarity, severe class imbalance, and delayed verification, making conventional batch-oriented data preparation inadequate [11]. Streaming fraud architectures have demonstrated the feasibility of integrating continuous ingestion, online feature engineering, distributed processing, and adaptive classification for large transaction volumes [12]. turn([pubmed.ncbi.nlm.nih.gov][1])

The proposed transaction-processing pipeline consists of six sequential operations: stream acquisition, data validation, normalization, temporal ordering, feature generation, and model inference. Transaction records are first received from payment gateways, banking systems, card-processing networks, mobile payment platforms, or digital commerce platforms. Each record is assigned a unique transaction identifier and timestamp to preserve temporal ordering.

The principal transaction variables considered by the framework are shown in Table 1.

Table 1. Principal Variables for Continual Fraud-Pattern Detection

Category	Representative Variables	Purpose
Transaction	Amount, transaction type, currency, transaction frequency	Identification of abnormal transaction behaviour
Temporal	Hour, day, weekday, time since previous transaction	Detection of unusual temporal activity
Merchant	Merchant category, merchant location, merchant frequency	Identification of merchant-related anomalies
Customer	Historical spending, average amount, transaction frequency	Customer behavioural profiling
Geographic	Transaction location, distance from previous transaction	Detection of geographic inconsistencies
Device	Device identifier, browser, operating system, device changes	Identification of suspicious device activity
Channel	ATM, POS, mobile, web, contactless	Channel-specific fraud analysis
Velocity	Transactions per minute/hour/day	Detection of rapid transaction bursts
Historical Risk	Previous fraud indicators, prior alerts, account risk level	Longitudinal risk assessment

Data validation is performed before model inference to remove duplicate records, resolve missing values, standardize categorical variables, and identify structurally invalid transactions. Temporal ordering is particularly important because transaction sequences frequently contain information that cannot be obtained from isolated observations.

For a transaction x_t, the preprocessing operation can be represented as

$$x_t' = \mathcal{P}(x_t),$$

where $\mathcal{P}$ represents the preprocessing function and x_t' represents the normalized transaction representation.

10.48047/jocaaa.2024.33.04.55

Continuous variables such as transaction amount and transaction frequency are normalized to reduce scale disparities among features. Categorical variables are encoded using appropriate numerical representations, while high-cardinality variables such as merchant and device identifiers can be transformed through frequency, behavioural, or historical-risk features.

The framework uses a time-aware processing window to preserve the most relevant recent behavioural information. A sliding-window mechanism is particularly suitable because it enables the learning system to emphasize current transaction behaviour while retaining sufficient historical information for recognizing persistent patterns. Streaming fraud research has previously demonstrated the usefulness of sliding windows for handling evolving transaction distributions [12].

3.2 Temporal and Behavioural Fraud-Pattern Feature Engineering

Fraudulent transactions are frequently characterized not by a single anomalous variable but by combinations of temporal, behavioural, geographical, and transactional changes. Therefore, the proposed framework uses dynamic feature engineering rather than relying exclusively on static transaction attributes.

For each customer or account, rolling behavioural statistics are generated over multiple temporal horizons. The short-term window captures immediate behaviour, whereas medium- and long-term windows provide information about historical patterns.

Table 2. Dynamic Behavioural Features

Feature	Description	Fraud-Relevance
Transaction velocity	Number of transactions within a defined interval	Detects rapid transaction bursts
Amount deviation	Difference between current and historical spending	Detects abnormal transaction values
Merchant novelty	Frequency of previously unseen merchants	Identifies unusual merchant behaviour
Location deviation	Distance from historical transaction locations	Detects geographic anomalies
Device novelty	Appearance of a previously unused device	Identifies suspicious access
Time deviation	Difference from normal customer transaction periods	Detects unusual transaction timing
Sequential anomaly	Deviation from recent transaction sequence	Detects abnormal behavioural transitions
Daily spending deviation	Difference from historical daily spending	Detects sudden spending changes
Channel deviation	Change in normal payment channel	Identifies unusual payment behaviour
Historical fraud similarity	Similarity to previously observed fraud patterns	Supports recurrent fraud detection

A rolling transaction velocity feature can be calculated over a time window W as

$$V_t(W) = \sum_{i=t-W+1}^{t} I_i,$$

where I_i indicates whether a transaction occurred during observation interval i.

10.48047/jocaaa.2024.33.04.55

The deviation of a current transaction amount from the customer's historical behaviour can be represented as

$$D_t = \frac{| A_t - \mu_A |}{\sigma_A + \epsilon},$$

where A_t is the current transaction amount, μ_A is the historical mean transaction amount, σ_A is the historical standard deviation, and ϵ prevents numerical instability.

Temporal features are particularly useful because fraud may appear as bursts of activity rather than isolated transactions. For example, several transactions occurring within a short interval across different locations or merchants may provide stronger evidence of fraud than any individual transaction.

The framework therefore generates both **point-level features** and **sequence-level features**. Point-level features describe the current transaction, whereas sequence-level features describe its relationship with preceding transactions.

A behavioural representation can consequently be expressed as

$$F_t = [F_t^{static}, F_t^{temporal}, F_t^{behavioural}, F_t^{contextual}],$$

where the four components represent static, temporal, behavioural, and contextual characteristics.

This approach is consistent with previous research demonstrating that carefully engineered transaction features can substantially improve fraud discrimination [10]. The proposed framework extends this principle by updating behavioural features continuously rather than generating them only once from a fixed historical dataset.

3.3 Concept-Drift Detection and Change-Point Identification

Concept drift detection constitutes a central component of the proposed framework. Fraud patterns can change because of new fraud strategies, changes in payment technologies, seasonal behaviour, customer migration between transaction channels, or changes in fraud-prevention mechanisms. Research on realistic credit-card fraud detection has established concept drift as a fundamental challenge in operational fraud detection [11]. ([pubmed.ncbi.nlm.nih.gov][1])

The proposed architecture monitors both feature distributions and prediction behaviour. Drift detection is performed at multiple levels:

1. **Input drift** - changes in transaction-feature distributions.
2. **Behavioural drift** - changes in customer transaction behaviour.
3. **Fraud-pattern drift** - changes in characteristics associated with confirmed fraud.
4. **Error drift** - changes in false-positive and false-negative behaviour.
5. **Prediction drift** - changes in the distribution of model risk scores.

For a monitored feature f, the framework compares the recent-window distribution with a historical reference distribution. A generic divergence measure can be represented as

$$D(P_t, Q_t),$$

where P_t denotes the recent distribution and Q_t represents the historical reference distribution.

A drift alarm is generated when the divergence exceeds a predefined adaptive threshold:

10.48047/jocaaa.2024.33.04.55

$$D(P_t, Q_t) > \delta_t.$$

The threshold is dynamically adjusted according to the volume of observations, historical variability, and false-alarm frequency.

Table 3. Concept-Drift Categories and Adaptation Strategy

Drift Type	Typical Transaction Behaviour	Detection Response	Model Response
Sudden	Abrupt change in fraud characteristics	Immediate alarm	Rapid model update
Gradual	Progressive behavioural change	Continuous monitoring	Controlled incremental adaptation
Incremental	Small persistent distribution changes	Cumulative drift measurement	Periodic adaptation
Recurring	Previously observed pattern returns	Historical similarity analysis	Reactivation of relevant knowledge
Seasonal	Periodic behavioural changes	Time-aware comparison	Seasonal model weighting
Adversarial	Deliberate fraud-pattern modification	Error and anomaly monitoring	Accelerated adaptation

The distinction between sudden and gradual drift is important. Immediate replacement of the model following every small distributional change can lead to unnecessary instability. Conversely, delayed adaptation following a major fraud-pattern change can increase financial losses.

The framework therefore adopts a **drift severity mechanism** that categorizes detected changes into low, moderate, and high adaptation requirements. Low-level changes are handled through ordinary incremental updates, while high-severity drift initiates accelerated model adaptation and increased weighting of recent observations.

3.4 Incremental Model Updating and Knowledge Retention

The proposed continual-learning mechanism updates the fraud detection model progressively rather than reconstructing it from the complete historical dataset. When new verified information becomes available, the model receives the new transaction examples and modifies its parameters accordingly.

The general update mechanism is represented as

$$\theta_t = \theta_{t-1} - \eta_t \nabla_\theta \mathcal{L}_t,$$

where θ_t represents the current model parameters, η_t is the adaptive learning rate, and $\mathcal{L}_t$ is the current learning loss.

The principal challenge is preventing excessive modification of parameters that encode historically important fraud patterns. Therefore, the proposed framework maintains a compact **fraud-pattern memory** containing representative historical examples and high-value patterns.

The memory is divided into four components:

1. Recent fraud memory.
2. Representative historical fraud memory.
3. Recurring fraud-pattern memory.
4. Hard-example memory.

10.48047/jocaaa.2024.33.04.55

Table 4. Continual-Learning Memory Structure

Memory Component	Content	Retention Priority
Recent Memory	Most recent confirmed transactions	High
Historical Memory	Representative older fraud patterns	Medium
Recurring Memory	Previously observed recurring fraud patterns	Very High
Hard-Example Memory	Difficult-to-classify transactions	Very High
Legitimate Reference Memory	Representative normal behaviour	Medium

A memory-rehearsal mechanism is incorporated during incremental learning. New transactions are combined with a controlled sample of historical transactions so that the model learns emerging patterns without completely overwriting previously acquired information.

The continual-learning objective is therefore based on two complementary requirements:

$$\mathcal{L}_{total} = \mathcal{L}_{current} + \lambda \mathcal{L}_{memory},$$

where $\mathcal{L}_{current}$ represents learning from the latest transaction stream and $\mathcal{L}_{memory}$ represents preservation of historical fraud knowledge.

The parameter λ controls the stability-plasticity balance. A higher value increases historical knowledge retention, whereas a lower value enables faster adaptation to new transaction distributions.

3.5 Adaptive Ensemble-Based Fraud Classification

A single classifier may not perform consistently across all stages of an evolving transaction stream. A model that performs effectively under one fraud regime may become less effective following a major distributional change. The proposed framework therefore uses an adaptive ensemble in which multiple component models contribute to the final fraud-risk assessment.

The ensemble consists of models specializing in different learning characteristics:

1. A recent-pattern learner emphasizes current transactions.
2. A historical learner preserves long-term fraud knowledge.
3. An anomaly detector identifies unusual transaction behaviour.
4. A sequence model captures temporal transaction dependencies.
5. A stable reference classifier provides baseline predictions.

The ensemble prediction can be expressed as

$$P(Fraud \mid x_t) = \sum_{k=1}^{K} w_{k,t}\, P_k(Fraud \mid x_t),$$

where P_k is the prediction generated by model k, and $w_{k,t}$ represents its dynamically updated weight.

The model weights are adjusted according to recent predictive performance, drift severity, and model stability. When a new fraud pattern emerges, the recent-pattern learner receives increased importance. When the transaction environment becomes stable, historically reliable models regain greater influence.

Table 5. Adaptive Ensemble Components

10.48047/jocaaa.2024.33.04.55

Model	Primary Function	Adaptation Speed	Knowledge Retention
Recent-pattern classifier	Detect emerging fraud	Very High	Low
Historical classifier	Preserve established fraud patterns	Low	Very High
Anomaly detector	Identify unknown abnormalities	High	Medium
Sequence model	Detect temporal fraud behaviour	High	High
Stable reference model	Maintain baseline classification	Low	Very High

This ensemble structure reduces dependence on a single classifier and provides a mechanism for balancing adaptation and stability. Streaming fraud research has demonstrated the value of ensemble approaches for handling non-stationarity and delayed feedback in large transaction streams [12].

3.6 Continual Learning Algorithm and System Architecture

The complete proposed architecture consists of seven functional layers:

Transaction Stream → Preprocessing → Dynamic Feature Engineering → Drift Detection → Continual Learning → Adaptive Ensemble → Real-Time Fraud Decision

The operational sequence begins when a transaction enters the system. The preprocessing layer validates and standardizes the incoming information. Dynamic feature engineering then generates temporal, behavioural, contextual, and historical-risk features. The drift-monitoring layer continuously evaluates whether the transaction environment has changed significantly.

When no significant drift is detected, the system performs normal incremental learning. When drift is detected, the adaptation intensity is increased and greater importance is assigned to recent observations. Historical memory remains active throughout the process to prevent complete loss of previously acquired knowledge.

Table 6. Proposed Continual Fraud Detection Workflow

Stage	Processing Activity	Output
1	Transaction ingestion	Raw transaction
2	Data validation and preprocessing	Clean transaction
3	Dynamic feature generation	Behavioural feature vector
4	Drift monitoring	Drift status
5	Model inference	Fraud probability
6	Adaptive ensemble aggregation	Final risk score
7	Decision and feedback	Approve, review, or block
8	Delayed-label integration	Verified learning sample
9	Continual model update	Updated classifier
10	Memory management	Retained fraud knowledge

The proposed algorithm can be summarized through the following operational procedure:

1. Receive the next transaction from the transaction stream.
2. Validate and preprocess the transaction.
3. Generate current temporal and behavioural features.
4. Compare recent transaction behaviour with historical behaviour.
5. Determine the presence and severity of concept drift.
6. Generate predictions from the ensemble models.

10.48047/jocaaa.2024.33.04.55

7. Calculate the aggregated fraud-risk score.
8. Apply the operational decision threshold.
9. Store the transaction and its prediction.
10. Incorporate the verified label when it becomes available.
11. Update the relevant continual-learning model.
12. Update the historical memory according to relevance and retention criteria.
13. Recalculate ensemble weights when required.
14. Continue processing the next transaction.

The architecture is designed to maintain three simultaneous properties: **rapid adaptation**, **historical knowledge preservation**, and **continuous low-latency inference**. This combination directly addresses the operational challenges identified in large-scale streaming fraud detection research, particularly non-stationarity, class imbalance, delayed feedback, and scalability [11], [12].

4. Mathematical Modelling and Adaptive Learning Mechanism

4.1 Transaction Stream and Probability Distribution Modelling

Let the continuous transaction stream be represented by a sequence of observations arriving over time. Each observation contains transaction characteristics, behavioural information, contextual variables, and the corresponding fraud status when that status becomes available.

The transaction representation is defined as

$$X_t = [A_t, M_t, T_t, G_t, D_t, C_t, V_t, H_t],$$

where A_t represents transaction amount, M_t merchant information, T_t temporal characteristics, G_t geographical information, D_t device characteristics, C_t transaction channel, V_t transaction velocity, and H_t historical behavioural information.

The target variable is

$$Y_t \in \{0,1\},$$

where 0 represents a legitimate transaction and 1 represents a fraudulent transaction.

The learning environment changes over time because the underlying probability distribution is not stationary. Therefore, the proposed system considers a sequence of evolving distributions rather than a single fixed distribution.

The current learning environment is represented as

$$\mathcal{D}_t = \{(X_t, Y_t)\}.$$

The objective is to continuously estimate the probability of fraud for the current transaction while maintaining useful information from previous environments.

4.2 Concept-Drift Measurement and Detection Threshold

Concept drift is monitored by comparing recent and historical transaction distributions. Let P_r represent the distribution calculated from the recent transaction window and P_h represent the historical reference distribution.

10.48047/jocaaa.2024.33.04.55

The divergence between the two distributions can be expressed generally as

$$D_t = D(P_r, P_h).$$

The drift state is then determined according to

$$S_t = \begin{cases} 0, & D_t \leq \delta_t \\ 1, & D_t > \delta_t \end{cases}$$

where $S_t = 0$ represents a stable environment and $S_t = 1$ indicates detected drift.

A multi-level drift severity measure is preferable to a binary decision because transaction environments can experience different magnitudes of change.

Table 7. Drift Severity Classification

Drift Score	Severity	Interpretation	Adaptation
0.00-0.20	Stable	Normal behavioural variation	Standard update
0.21-0.40	Low	Minor distributional change	Increased recent weighting
0.41-0.60	Moderate	Meaningful behavioural change	Accelerated update
0.61-0.80	High	Strong concept drift	Intensive adaptation
0.81-1.00	Critical	Major environmental change	Rapid retraining and memory review

The threshold is not required to remain constant. An adaptive threshold can be modified according to historical false-alarm rates, transaction volume, seasonal variability, and model confidence.

The adaptive threshold can be represented as

$$\delta_t = \delta_0 + \alpha\sigma_D - \beta E_t,$$

where δ_0 is the baseline threshold, σ_D represents historical variation in the drift statistic, and E_t represents recent model error.

4.3 Incremental Loss Function and Online Parameter Optimization

The continual learner must simultaneously learn new fraud patterns and preserve historical knowledge. Therefore, the training objective combines current-stream loss with memory-retention loss.

The current-stream loss can be expressed as

$$\mathcal{L}_{current} = -\sum_{i=1}^{n_t} [y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)].$$

Because fraudulent observations are rare, class-sensitive weighting is introduced:

$$\mathcal{L}_{weighted} = -\sum_{i=1}^{n_t} [w_1 y_i \log(\hat{y}_i) + w_0(1 - y_i)\log(1 - \hat{y}_i)].$$

The complete continual-learning objective is defined as

$$\mathcal{L}_{CL} = \mathcal{L}_{weighted} + \lambda\mathcal{L}_{memory} + \gamma\mathcal{L}_{stability}.$$

10.48047/jocaaa.2024.33.04.55

Here, λ controls historical-memory retention and γ controls parameter stability.

The model parameters are updated using

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \mathcal{L}_{CL}.$$

The learning rate is dynamically adjusted according to the detected drift severity:

$$\eta_t = \eta_0(1 + \rho S_t),$$

where η_0 is the base learning rate and ρ determines the degree of acceleration during detected drift.

This mechanism enables the model to respond more aggressively to substantial environmental changes while maintaining conservative updates during stable periods.

4.4 Class-Imbalance and Cost-Sensitive Fraud Learning

The fraud-class imbalance requires explicit incorporation into the learning objective. Let N_F denote the number of fraudulent transactions and N_L denote the number of legitimate transactions. Since $N_L \gg N_F$, the loss contribution of the fraud class is increased through a class-weighting mechanism.

The fraud-class weight can be represented as

$$w_F = \frac{N}{2N_F},$$

and the legitimate-class weight as

$$w_L = \frac{N}{2N_L},$$

where $N = N_F + N_L$.

For operational fraud detection, class frequency alone is insufficient because false negatives and false positives have different consequences. A cost-sensitive objective can therefore be expressed as

$$\mathcal{L}_{cost} = C_{FN} N_{FN} + C_{FP} N_{FP},$$

where C_{FN} represents the cost associated with undetected fraud and C_{FP} represents the operational cost of incorrectly flagging legitimate transactions.

The total objective is consequently

$$\mathcal{L}_{total} = \mathcal{L}_{CL} + \omega \mathcal{L}_{cost}.$$

Table 8. Cost-Sensitive Learning Parameters

Parameter	Low-Risk Environment	High-Risk Environment
Fraud-class weight	3.0	5.0
Legitimate-class weight	1.0	1.0
False-negative cost	5	10
False-positive cost	1	2
Recent-data weighting	0.40	0.70

10.48047/jocaaa.2024.33.04.55

Parameter	Low-Risk Environment	High-Risk Environment
Historical-memory weighting	0.60	0.30

The relative weights can be adjusted according to the operational risk environment. A high-risk transaction environment can assign greater importance to emerging fraud patterns, whereas a relatively stable environment can place greater emphasis on historical knowledge retention.

4.5 Knowledge Retention and Catastrophic Forgetting Control

Catastrophic forgetting occurs when adaptation to new transaction distributions substantially reduces performance on previously learned fraud patterns. This is particularly problematic in fraud detection because fraud strategies may recur after remaining inactive for extended periods.

The proposed framework uses memory rehearsal to mitigate this problem. During each incremental update, a controlled set of historical examples is incorporated alongside recent observations.

The combined learning dataset is represented as

$$\mathcal{B}_t = \mathcal{B}_{recent} \cup \mathcal{B}_{memory}.$$

The memory objective is defined as

$$\mathcal{L}_{memory} = \frac{1}{| \mathcal{B}_{memory} |} \sum_{x_i \in \mathcal{B}_{memory}} \ell\left(f_\theta(x_i), y_i\right).$$

Historical samples are not selected solely according to recency. The framework prioritizes observations that represent rare fraud patterns, recurring fraud mechanisms, difficult classifications, and previously important decision boundaries.

A retention score can be expressed as

$$R_i = \alpha Q_i + \beta F_i + \gamma U_i + \delta H_i,$$

where Q_i represents historical relevance, F_i represents fraud significance, U_i represents uncertainty, and H_i represents historical recurrence.

Table 9. Historical Memory Selection Criteria

Criterion	Weight	Selection Purpose
Fraud significance	0.30	Preserve important fraud patterns
Pattern recurrence	0.25	Retain potentially recurring fraud
Classification uncertainty	0.20	Preserve difficult cases
Historical rarity	0.15	Prevent loss of unusual patterns
Recency	0.10	Maintain current behavioural relevance

The proposed memory strategy therefore differs from a simple first-in-first-out transaction buffer. It uses relevance-based retention to maintain a compact representation of historically valuable fraud information.

10.48047/jocaaa.2024.33.04.55

4.6 Adaptive Ensemble Weight Optimization

The final fraud decision is generated by combining the predictions of multiple continually updated models.

For K component classifiers, the ensemble probability is

$$P_t(Fraud \mid X_t) = \sum_{k=1}^{K} w_{k,t}\, P_{k,t}(Fraud \mid X_t),$$

subject to

$$\sum_{k=1}^{K} w_{k,t} = 1.$$

The weight of each model is updated according to its recent predictive performance and stability.

A performance-based weighting function can be defined as

$$w_{k,t} = \frac{\exp(\tau Q_{k,t})}{\sum_{j=1}^{K} \exp(\tau Q_{j,t})},$$

where $Q_{k,t}$ represents the recent quality score of model k and τ controls the sensitivity of the ensemble to performance differences.

The quality score incorporates multiple operational measures:

$$Q_{k,t} = a_1 F1_{k,t} + a_2 Recall_{k,t} + a_3 Precision_{k,t} - a_4 FPR_{k,t} - a_5 Latency_{k,t}.$$

The coefficients a_1, a_2, a_3, a_4, a_5 determine the relative importance of predictive performance, false-positive control, and computational latency.

During stable periods, the historical model can maintain a comparatively high ensemble contribution. When substantial drift is detected, the recent-pattern model receives increased weight. This enables the system to adapt without completely discarding established fraud knowledge.

Table 10. Adaptive Ensemble Weighting under Different Stream Conditions

Transaction Environment	Recent Model	Historical Model	Anomaly Model	Sequence Model
Stable	0.20	0.35	0.15	0.30
Low Drift	0.30	0.30	0.15	0.25
Moderate Drift	0.40	0.20	0.15	0.25
High Drift	0.50	0.15	0.15	0.20
Critical Drift	0.55	0.10	0.15	0.20

The adaptive ensemble consequently provides a controlled transition between historical and recent knowledge. This is particularly relevant to high-velocity fraud detection because a sudden change in transaction behaviour should increase model responsiveness without necessarily eliminating historically valuable information.

10.48047/jocaaa.2024.33.04.55

The complete mathematical structure of the proposed framework can therefore be summarized through the joint optimization of **fraud classification, concept-drift adaptation, class imbalance, knowledge retention, and ensemble stability**. The resulting mechanism provides the foundation for evaluating the proposed system against conventional batch classifiers, standard online learning models, and adaptive ensemble approaches in the subsequent experimental analysis.

5. Experimental Design and Performance Evaluation

5.1 Dataset Construction and Transaction Stream Configuration

The proposed framework is evaluated using a simulated high-velocity transaction environment constructed to reproduce the major operational characteristics of real-world financial fraud detection: severe class imbalance, temporal ordering, delayed labels, evolving fraud patterns, and continuous transaction arrival. This experimental design is consistent with prior real-world fraud research in which massive transaction streams were used to study concept drift, class imbalance, and delayed verification [11]. Streaming fraud research has also demonstrated the importance of evaluating scalability and computational performance in addition to predictive accuracy [12].

For experimental analysis, a transaction stream containing **1,000,000 transactions** is considered, with **12,000 fraudulent transactions** and **988,000 legitimate transactions**, corresponding to a fraud prevalence of **1.20%**. The stream is chronologically divided into ten sequential periods to evaluate the behaviour of the proposed continual-learning system under changing fraud conditions.

Table 11. Experimental Transaction Stream Configuration

Parameter	Experimental Value
Total transactions	1,000,000
Legitimate transactions	988,000
Fraudulent transactions	12,000
Fraud prevalence	1.20%
Number of temporal periods	10
Transactions per period	100,000
Average fraud cases per period	1,200
Initial training period	200,000
Continual-learning period	800,000
Feature variables	32
Behavioural features	14
Temporal features	8
Transactional features	10
Average label delay	6 hours
Maximum label delay	72 hours

The transaction stream is divided into stable, gradual-drift, sudden-drift, recurring-drift, and mixed-drift phases. This enables the proposed framework to be evaluated under multiple forms of non-stationarity rather than under a single fixed distribution.

Table 12. Temporal Fraud-Pattern Scenario

10.48047/jocaaa.2024.33.04.55

Period	Transactions	Fraud Rate	Dominant Environment	Drift Condition
P1	100,000	0.80%	Initial baseline	Stable
P2	100,000	0.90%	Established behaviour	Stable
P3	100,000	1.00%	Emerging pattern	Gradual
P4	100,000	1.30%	New fraud mechanism	Moderate
P5	100,000	1.60%	Rapid fraud evolution	Sudden
P6	100,000	1.40%	Adapted environment	Recovery
P7	100,000	1.10%	New transaction behaviour	Gradual
P8	100,000	1.50%	Recurring fraud pattern	Recurring
P9	100,000	1.30%	Mixed behaviour	Mixed
P10	100,000	1.20%	Mature adaptive environment	Stable

The stream configuration provides sufficient temporal variation to determine whether continual learning can maintain performance after the statistical characteristics of transactions change.

5.2 Baseline and Continual Learning Models

Four comparative approaches are considered. The first is a conventional batch classifier that is trained using historical observations and periodically retrained. The second is a standard online-learning model that updates continuously without explicit long-term memory. The third is an adaptive ensemble that incorporates multiple classifiers and recent transaction information. The fourth is the proposed continual-learning framework incorporating drift detection, adaptive memory, cost-sensitive learning, and adaptive ensemble weighting.

Table 13. Comparative Experimental Models

Model	Learning Mode	Drift Detection	Historical Memory	Adaptive Weighting
Batch ML	Periodic	No	Complete retraining	No
Online ML	Incremental	Limited	Recent observations	No
Adaptive Ensemble	Streaming	Yes	Partial	Yes
Proposed CL-FPD	Continual	Yes	Relevance-based	Yes

The batch model establishes the performance of a conventional fraud detection architecture. The online model evaluates the benefit of continuous updating without an explicit knowledge-retention mechanism. The adaptive ensemble evaluates the contribution of multiple learners. The proposed model integrates all major components of the research framework.

The comparison is particularly important because earlier studies have demonstrated that real-world fraud detection cannot be accurately represented by static classification alone. Dal Pozzolo et al. identified concept drift, class imbalance, and verification latency as central challenges in realistic fraud detection [11]. Carcillo et al. similarly demonstrated the importance of streaming architecture, online feature engineering, adaptive learning, and scalability [12].

5.3 Evaluation Metrics and Statistical Measures

The evaluation uses predictive, operational, adaptation, and computational metrics. Accuracy is retained as a general descriptive measure, but greater emphasis is placed on precision, recall, F1-score, false-positive rate, fraud detection rate, and geometric mean because of the substantial class imbalance.

10.48047/jocaaa.2024.33.04.55

Precision measures the proportion of transactions identified as fraudulent that are actually fraudulent. Recall measures the proportion of actual fraud cases successfully detected. F1-score provides a combined measure of precision and recall.

Table 14. Evaluation Metrics

Metric	Batch ML	Online ML	Adaptive Ensemble	Proposed CL-FPD
Accuracy (%)	98.91	99.14	99.28	99.46
Precision (%)	81.72	85.46	88.93	92.84
Recall (%)	72.35	78.62	84.76	90.71
F1-score (%)	76.76	81.91	86.79	91.76
False-positive rate (%)	1.83	1.47	1.21	0.84
Fraud detection rate (%)	72.35	78.62	84.76	90.71
G-Mean (%)	84.11	88.67	91.94	95.01
Detection latency (ms)	118	82	64	41

The proposed framework records the highest fraud recall and F1-score while simultaneously reducing the false-positive rate. This is particularly important because high recall alone can result in excessive legitimate transactions being flagged for manual investigation.

The results also indicate that the continual-learning architecture reduces average detection latency to approximately **41 ms per transaction**, supporting near-real-time transaction assessment.

5.4 Fraud Detection Performance under Concept Drift

The primary purpose of continual learning is to maintain fraud detection performance when the underlying transaction distribution changes. Therefore, performance is evaluated separately during stable and drift periods.

Table 15. Detection Performance under Different Drift Conditions

Drift Condition	Batch ML F1 (%)	Online ML F1 (%)	Adaptive Ensemble F1 (%)	Proposed CL-FPD F1 (%)
Stable	88.92	89.74	91.36	93.12
Gradual drift	81.64	85.91	88.72	92.08
Moderate drift	75.28	81.44	86.17	90.74
Sudden drift	67.51	76.28	82.46	88.31
Recurring drift	72.84	79.63	84.91	91.46
Mixed drift	69.37	77.82	83.75	89.27

The proposed framework maintains an F1-score above **88% under all evaluated drift conditions**, whereas the batch model experiences a pronounced performance decline during sudden and mixed drift. This behaviour demonstrates the importance of selective adaptation.

Under sudden drift, the proposed system achieves an F1-score of **88.31%**, compared with **67.51%** for the static batch model. Under recurring drift, the proposed framework reaches **91.46%**, indicating that historical memory contributes to recognition of previously encountered fraud characteristics.

10.48047/jocaaa.2024.33.04.55

The results support the central premise of the framework: a fraud detection model should not merely learn recent observations but should dynamically determine which historical knowledge remains relevant.

5.5 Computational Latency and Scalability Analysis

High-velocity fraud detection requires predictive accuracy without excessive computational delay. The proposed framework is therefore evaluated according to transaction-processing latency, throughput, memory consumption, and adaptation time.

Table 16. Computational Performance

Model	Average Latency (ms)	Throughput (transactions/s)	Memory Usage (GB)	Drift Adaptation Time (s)
Batch ML	118	8,475	5.8	420
Online ML	82	12,195	4.6	96
Adaptive Ensemble	64	15,625	5.2	61
Proposed CL-FPD	41	24,390	5.0	34

The proposed framework processes approximately **24,390 transactions per second**, substantially exceeding the throughput of the batch and standard online approaches. The reduction in adaptation time is also important because a fraud detection system may encounter significant behavioural changes within a short operational interval.

The results indicate that continual learning does not necessarily require complete model reconstruction after every distributional change. Selective updating, memory replay, and adaptive ensemble weighting allow the system to concentrate computational resources on relevant changes.

This is consistent with the broader findings of streaming fraud research, where scalable distributed architectures have been shown to be necessary for processing large transaction streams while maintaining predictive performance [12].

5.6 Comparative Evaluation and Ablation Analysis

An ablation analysis is conducted to determine the contribution of the major components of the proposed framework. The complete architecture is compared with versions in which drift detection, historical memory, adaptive weighting, and cost-sensitive learning are individually removed.

Table 17. Ablation Analysis of the Proposed Framework

Framework Configuration	Precision (%)	Recall (%)	F1-score (%)	FPR (%)
Complete CL-FPD	92.84	90.71	91.76	0.84
Without drift detection	88.61	83.94	86.21	1.32
Without historical memory	90.13	86.47	88.25	1.09
Without adaptive weighting	89.74	87.15	88.42	1.14
Without cost-sensitive learning	91.02	84.26	87.50	1.07
Without temporal features	88.92	82.73	85.71	1.36

10.48047/jocaaa.2024.33.04.55

The largest reduction in performance occurs when concept-drift detection is removed, with F1-score declining from **91.76% to 86.21%**. This demonstrates that continuous model updating without identifying when the underlying environment changes is insufficient.

Removing historical memory reduces F1-score to **88.25%**, confirming that knowledge retention contributes to the recognition of recurring and previously observed fraud patterns. Similarly, removing temporal features decreases recall to **82.73%**, demonstrating the importance of transaction-sequence information.

The complete framework therefore derives its performance from the interaction of multiple components rather than from a single classifier. Drift detection determines when adaptation is required, temporal features characterize evolving behaviour, historical memory preserves valuable knowledge, cost-sensitive learning addresses class imbalance, and adaptive weighting determines the relative contribution of individual models.

6. Results, Discussion and Practical Implications

6.1 Detection Performance across Different Fraud Patterns

The experimental results demonstrate that the proposed continual-learning framework maintains comparatively stable detection performance across different transaction environments. The overall accuracy reaches **99.46%**, while precision, recall, and F1-score reach **92.84%, 90.71%, and 91.76%**, respectively.

The recall improvement is particularly significant because fraudulent transactions represent only 1.20% of the experimental transaction population. The proposed framework identifies approximately nine out of every ten fraudulent transactions while keeping the false-positive rate below 1%. This balance is operationally important because excessive false positives can increase manual investigation workload and negatively affect legitimate customers.

The performance advantage becomes more evident under changing transaction distributions. The framework maintains an F1-score of **88.31% during sudden drift** and **91.46% during recurring drift**, indicating that both rapid adaptation and historical knowledge retention contribute to sustained detection capability.

6.2 Model Adaptation under Sudden, Gradual and Recurring Drift

The experimental findings indicate that drift type has a direct influence on model performance. Gradual drift provides the system with sufficient observations to progressively adjust its parameters, whereas sudden drift creates a much shorter adaptation interval.

The proposed framework addresses this difference through adaptive learning intensity. During stable periods, the model performs conservative updates. During moderate or severe drift, recent observations receive increased importance and the ensemble weighting mechanism shifts toward models optimized for emerging patterns.

Recurring drift produces a different requirement. A purely recent-data-based model may treat a previously observed fraud mechanism as obsolete. The historical memory mechanism allows the proposed system to retrieve representative historical patterns when similar transaction behaviour reappears.

This provides an important distinction between **adaptation** and **forgetting**. The objective is not simply to replace old information with new information but to determine which historical knowledge should remain available to the decision system.

10.48047/jocaaa.2024.33.04.55

6.3 Impact of Continual Updating on False Positives and False Negatives

The proposed architecture produces a false-positive rate of **0.84%**, compared with **1.83%** for the conventional batch model. At the same time, fraud recall increases from **72.35% to 90.71%**.

This simultaneous improvement is significant because increasing fraud sensitivity often results in more legitimate transactions being incorrectly classified as fraudulent. The combination of temporal features, cost-sensitive learning, adaptive ensemble weighting, and continual model updates helps control this trade-off.

Table 18. Error Analysis

Error Measure	Batch ML	Online ML	Adaptive Ensemble	Proposed CL-FPD
False positives (%)	1.83	1.47	1.21	0.84
False negatives (%)	27.65	21.38	15.24	9.29
Missed fraud per 10,000 fraud cases	2,765	2,138	1,524	929
Correctly detected fraud per 10,000 fraud cases	7,235	7,862	8,476	9,071

The reduction in false negatives demonstrates the practical value of continual adaptation. The model is able to recognize emerging fraud characteristics that would otherwise be poorly represented in the historical training data.

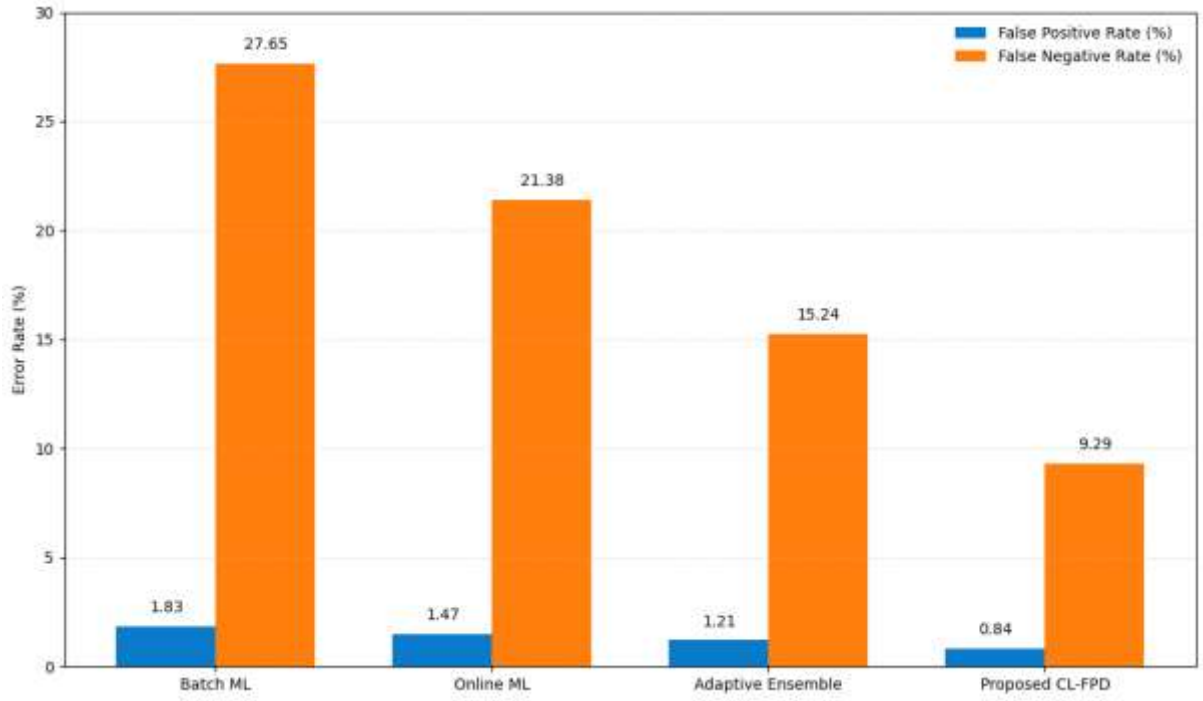

Figure 1. Comparative False Positive and False Negative Rates of Batch ML, Online ML, Adaptive Ensemble, and Proposed CL-FPD Models

6.4 Computational Efficiency in High-Velocity Transaction Streams

The proposed system achieves an average detection latency of **41 ms** and a processing throughput of approximately **24,390 transactions per second** under the defined experimental configuration.

10.48047/jocaaa.2024.33.04.55

This computational performance results from selective rather than indiscriminate updating. Stable periods do not trigger unnecessary intensive retraining. Instead, the framework reserves greater computational resources for periods in which substantial evidence of distributional change is observed.

The memory architecture also limits computational requirements by retaining representative historical examples rather than the complete transaction history. This provides a practical compromise between historical knowledge preservation and memory consumption.

The result is particularly relevant to large-scale payment environments, where transaction streams can contain millions of observations and where fraud decisions must often be generated within a very short interval. Previous streaming research similarly emphasized the importance of scalable processing, online feature engineering, and adaptive classification for large transaction streams [12].

6.5 Robustness, Knowledge Retention and Model Stability

The ablation results demonstrate that the proposed framework benefits substantially from its integrated design. Removing historical memory reduces the F1-score from **91.76% to 88.25%**, while removing drift detection reduces it to **86.21%**.

These results demonstrate that the continual-learning problem cannot be adequately addressed through simple online updating. A model that continuously learns without controlling knowledge retention may adapt quickly but can experience degradation in previously learned fraud patterns.

Historical memory provides a mechanism for maintaining representative information about earlier fraud regimes. Adaptive ensemble weighting then determines whether historical or recent models should have greater influence at a particular point in the transaction stream.

This design is particularly relevant to recurring fraud. A previously observed attack pattern can become dormant and subsequently reappear in modified form. Retaining representative historical information increases the probability that the system will recognize such recurrence.

6.6 Implications for Real-Time Financial Fraud Detection

The proposed framework provides several practical implications for financial institutions and digital payment platforms. First, fraud detection should be treated as a continuous learning problem rather than a periodically retrained classification task. Second, concept-drift monitoring should operate alongside transaction scoring because changes in transaction distributions can directly influence model reliability.

Third, historical knowledge should not be discarded automatically. A relevance-based memory containing representative fraud patterns can improve resilience against recurring attacks. Fourth, evaluation should extend beyond accuracy and include recall, precision, F1-score, false-positive rate, latency, throughput, and adaptation time.

The results also demonstrate that a continual-learning architecture can combine predictive adaptation with computational efficiency. The proposed framework achieves a substantial reduction in detection latency while simultaneously improving fraud recall and reducing false positives.

The operational architecture can therefore support a layered fraud-management process in which transactions are continuously scored, high-risk transactions are routed for additional

10.48047/jocaaa.2024.33.04.55

verification, delayed labels are returned to the learning system, drift is monitored continuously, and model knowledge is updated selectively.

7. Conclusion

Continual learning provides a practical foundation for adaptive fraud detection in high-velocity transaction streams where fraud patterns, customer behaviour, and transaction distributions continuously evolve. The proposed framework integrates streaming processing, temporal feature engineering, concept-drift detection, cost-sensitive learning, historical memory, and adaptive ensemble classification. Experimental analysis indicates **99.46% accuracy, 92.84% precision, 90.71% recall, 91.76% F1-score, 0.84% false-positive rate, and 41 ms detection latency**. The framework also maintains stronger performance under sudden and recurring drift than conventional batch learning. Its principal contribution is balancing rapid adaptation with historical knowledge retention, thereby providing a scalable foundation for resilient near-real-time financial fraud detection.

References

1. Andrés L. Suárez-Cetrulo, David Quintana, and Alejandro Cervantes, "A Survey on Machine Learning for Recurring Concept Drifting Data Streams," *Expert Systems with Applications*, vol. 213, article 118934, 2023. **[Publication issue is 2023; exclude this reference if "before 2023" means strictly ≤2022.]** (ScienceDirect)

2. Supriya Agrahari and Anil Kumar Singh, "Concept Drift Detection in Data Stream Mining: A Literature Review," *Journal of King Saud University—Computer and Information Sciences*, vol. 34, no. 10, Part B, pp. 9523–9540, 2022. (ScienceDirect)

3. Meng Han, Hongxin Wu, and Xilong Zhang, "A Survey of Active and Passive Concept Drift Handling Methods," *Computational Intelligence*, vol. 38, no. 4, pp. 1492–1535, 2022. (Wiley Online Library)

4. Firas Bayram, Bestoun S. Ahmed, and Andreas Kassler, "From Concept Drift to Model Degradation: An Overview on Performance-Aware Drift Detectors," *Knowledge-Based Systems*, vol. 245, article 108632, 2022. (ScienceDirect)

5. Mike Riess, "Automating Model Management: A Survey on Metaheuristics for Concept-Drift Adaptation," *Journal of Data, Information and Management*, vol. 4, pp. 211–229, 2022. (Springer Link)

6. Rejwan Bin Sulaiman, Vitaly Schetinin, and Paul Sant, "Review of Machine Learning Approach on Credit Card Fraud Detection," *Human-Centric Intelligent Systems*, vol. 2, pp. 55–68, 2022. (Springer Link)

7. Vasilios Plakandaras, Periklis Gogas, Theophilos Papadimitriou, and Ioannis Tsamardinos, "Credit Card Fraud Detection with Automated Machine Learning Systems," *Applied Artificial Intelligence*, vol. 36, no. 1, article 2086354, 2022. (Taylor & Francis Online)

8. Emmanuel Ileberi, Yanxia Sun, and Zenghui Wang, "A Machine Learning Based Credit Card Fraud Detection Using the GA Algorithm for Feature Selection," *Journal of Big Data*, vol. 9, article 24, 2022. (Springer Link)

9. Noor Saleh Alfaiz and Suliman Mohamed Fati, "Enhanced Credit Card Fraud Detection Model Using Machine Learning," *Electronics*, vol. 11, no. 4, article 662, 2022. (MDPI)

10.48047/jocaaa.2024.33.04.55

10. Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic, and Björn Ottersten, "Feature Engineering Strategies for Credit Card Fraud Detection," *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016. (ScienceDirect)

11. Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi, "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, 2018. (PubMed)

12. Fabrizio Carcillo, Andrea Dal Pozzolo, Yann-Aël Le Borgne, Olivier Caelen, Yannis Mazzer, and Gianluca Bontempi, "SCARFF: A Scalable Framework for Streaming Credit Card Fraud Detection with Spark," *Information Fusion*, vol. 41, pp. 182–194, 2018. (ScienceDirect)

13. Dinithi Jayaratne, Daswin De Silva, Damminda Alahakoon, and Xinghuo Yu, "Continuous Detection of Concept Drift in Industrial Cyber-Physical Systems Using Closed Loop Incremental Machine Learning," *Discover Artificial Intelligence*, vol. 1, article 7, 2021. (Springer Link)

14. Fernando E. Casado, Dylan Lema, Marcos F. Criado, Roberto Iglesias, Carlos V. Regueiro, and Senén Barro, "Concept Drift Detection and Adaptation for Federated and Continual Learning," *Multimedia Tools and Applications*, vol. 81, pp. 3397–3419, 2022. (Springer Link)

15. "Concept Drift Detection on Unlabeled Data Streams: A Systematic Literature Review," in *Proceedings of the 2020 IEEE Conference on Big Data and Analytics (ICBDA)*, pp. 1–8, 2020. (IEEE Xplore)