Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Adaptive Edge-Cloud Intelligence for Low-Latency Anomaly Detection in Digital Payment Ecosystems

Mahatma Reddy Marri

Advanced Systems Administrator, Independent Researcher, Texas, USA

ARTICLEINFO

Received: 01 Feb 2024

Revised: 19 March 2024

Accepted: 29 March 2024

ABSTRACT

Digital payment systems process billions of transactions every day on a set of diverse infrastructure that ranges from resource-constrained point-of-sale terminals, near-edge gateways, regional cloud nodes, to central orchestration clusters. Round-trip cloud-based fraud detection systems are 100 to 200 milliseconds, which isn't feasible for on-demand payment authorisation, and pure on-device fraud detection systems are simply insufficient to represent the growing complexity and sophistication of fraud patterns. We introduce a novel framework called AEC-AD (Adaptive Edge-Cloud Anomaly Detector), which dynamically splits the inference between edge and cloud by leveraging a learned offloading policy across the edge-cloud continuum, includes lightweight streaming models on the edge, periodically synchronised teacher networks on cloud using knowledge distillation, and integrates an online concept drift detector to refresh the model proactively when the payment behaviour changes. AEC-AD achieves an AUC-ROC of 0.953 and an F1 score of 0.924 with an inference latency of 23.4ms at the 99th percentile performance, a 4.9× latency improvement over the cloud-only baselines and an 87% accuracy gain towards the prohibitively-high accuracy ceiling. A two-way ANOVA for bandwidth and system-load conditions shows that both bandwidth and load have significant main effects ($\eta^2 = 0.341$ and 0.202, respectively). Pairwise Wilcoxon signed-rank tests are used to determine the statistically significant advantage over six competing baselines ($p < 0.05$, $r \geq 0.442$, rank-biserial). The six incremental configurations are used in an ablation study to determine the contribution of each architectural component. The results of these findings make AEC-AD a deployable, regulation aware, intelligence architecture for the next generation payment security infrastructures.

Keywords: edge computing, anomaly detection, payment fraud, knowledge distillation, adaptive offloading, concept drift, low-latency inference, digital payments

1. INTRODUCTION

In 2023, the volume of transactions processed in the global digital payments market surpassed USD 8.49 trillion, with financial institutions and merchants losing nearly USD 0.053 billion annually to payment fraud (Kumar et al., 2023). This has significantly changed the fraud threat landscape as adversaries are now attacking accounts at machine speed with account takeover, synthetic identity fraud and card-present attacks taking advantage of the time lag between when the payment is made and when the payment is authorised (Alarab et al., 2022).

However, traditional cloud-based anomaly detection systems send every transaction to a distant inference server, resulting in 100-200ms latency for the deep neural network to achieve high accuracy in detecting anomalies. This latency budget is frequently breached during peak load or when connectivity is not optimal (Zhang et al., 2022). The deployment of inference at the edge, near the point of transaction origination (POS terminals, payment gateways and regional switching centres), provides a way to make very local decisions within a sub-10 millisecond timeframe but limits model complexity to what can fit within the memory and compute envelopes of embedded hardware (Wang et al., 2023).

This design space is characterized by the three following tensions. First, there is the accuracy-latency trade-off: more accurate models are more computation-heavy, which can't be done within the edge latency budgets. Second,

the static-model problem: the distributions of payment behaviour change over time as a result of seasonal influences, merchant category evolution and adversarial adaptation, leading to static on-device models that silently degrade over time. Third, the offloading decision problem: sending everything to the cloud without any consideration of bandwidth and falling short of latency requirements means that the accuracy of decisions on more complex fraud patterns that are not easy to predict and therefore require global context is sacrificed.

This paper brings all the above tensions into focus using the Adaptive Edge-Cloud Anomaly Detector (AEC-AD) with the following major contributions:

(2) Dynamic routing of every transaction to a processing tier (on-device, near-edge, cloud) based on a lightweight confidence estimator, features of a transaction to be processed and real-time network conditions, with a per-transaction service-level agreement (SLA) for transaction latency.

A hierarchical knowledge distillation protocol that periodically retrains the cloud teacher network to reduce behaviour knowledge to a small edge student model, so that the edge models can leverage the global fraud intelligence without transfer of raw data.

(3) A concept-drift detector (CDD) which is performed online on the rolling empirical fraud rate using the Page-Hinkley test, to trigger targeted model refresh at affected edge nodes without using the global model for retraining.

A thorough statistical analysis (two-way ANOVA with partial eta-squared effect sizes, Wilcoxon signed-rank tests with Bonferroni correction, bootstrap confidence intervals [B = 10,000] and McNemar tests for disagreement at the transaction level) that revealed statistically significant superiority over 7 baselines across 6 disparate payment datasets.

2. RELATED WORK

2.1 Anomaly Detection in Payment Systems

Traditional methods for early payment fraud detection were based on rule-based engines and classical machine learning models such as decision trees, logistic regression, and support vector machines, which were trained with handcrafted features (Bhattacharyya et al., 2021). They are highly dependent on skilled care and maintenance, and do not address multivariate interaction patterns. To overcome representational challenges, deep learning techniques have been employed to model temporal transaction sequences using recurrent architectures such as LSTM and GRU (Jiang et al., 2022), spatial co-occurrence patterns, across merchant categories, via convolutional networks (Sun et al., 2023), and long-range temporal dependence, across transaction histories, using transformer-based models (Chen et al., 2023). Moreover, the relations among entities capture ring fraud and money laundering networks, which are learnt by graph neural networks (Liu et al., 2022). These architectures, however, are optimized for batch training in the central system, and for inference to be deployed in the cloud, which is not suitable for latency-critical payment endpoints.

2.2 Edge Computing for Financial Applications

While the integration of edge intelligence in IoT and telecommunications continues to grow in popularity, its use in financial fraud detection is still in its early stages (Wang et al., 2023). Xie et al. (2022) introduced a pre-filed fraud-scoring system at the Point of Sale (POS) terminal, which flags transactions as low-risk and sends any suspicious transactions to the cloud for further review, thereby cutting down on cloud API calls by 64% at the POS terminal. Li et al. (2022) studied energy-accuracy trade-offs in edge deployment of quantised classifiers for fraud. They found that 8-bit Integer quantisation can achieve 97.8% of floating-point F1 score with a 4× memory reduction. Some researchers, including Alarab et al. (2022), tried to evaluate on-device graph neural networks for crypto payment fraud, but found these networks to be too slow to operate on ARM Cortex-M devices, calling for adaptive on-device offloading. All of these papers do not consider the complete spectrum of edge-cloud continuum or dynamic adaptation needed in the presence of concept drift.

2.3 Adaptive Inference and Offloading

In computer vision and NLP, adaptive neural network inference – early exit, dynamic depth selection and confidence-based routing – has been widely investigated (Teerapittayanon et al., 2021). Neurosymbolic approaches

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

are a combination of symbolic rule engines and neural confidence scores to make routing decisions (Zhang et al., 2022). To optimize computation delay-accuracy tradeoff, reinforcement learning policies have been used for mobile-cloud offloading (Kumar et al., 2023). Hu et al. (2022) introduced a multi-exit transformer (MET) for real-time video analytics, with the intention of providing low latency services to early exits. The translation of these concepts to payment fraud must be done with the domain-specific confidence calibration, since the base rates of fraud are much lower (between 0.1 and 0.5%) than the typical vision base rates, resulting in a severe class imbalance that makes the standard softmax confidence estimates inaccurate.

2.4 Knowledge Distillation for Edge Deployment

For edge deployment, knowledge distillation (Hinton et al., 2015) has been heavily used to shrink large models. To achieve a more robust representation from teacher networks than soft label distillation can, Romero et al. (2021) proposed intermediate-layer feature matching, which allows student networks to learn powerful representations from teacher networks. Lin et al. (2022) showed that, in the federated setting, there is a communication-efficient alternative to gradient aggregation: ensemble distillation. For federated financial models, Yao et al. (2023) used distillation to reduce cloud-trained fraud detectors to deploy them on payment terminals with 256 MB of RAM requirements. AEC-AD is an extension of this paradigm to the edge-cloud continuum, where the distillation happens hierarchically across several tiers and there are different compression targets at each tier.

2.5 Concept Drift in Financial Data

Payment transaction distributions are non-stationary: consumer spending patterns change seasonally, fraud tactics change in response to their detection, and a macroeconomic event can abruptly change the distribution of payments (Sun et al., 2023). Concept drift detection methods are ways of tracking either performance statistics or data distribution metrics to trigger model updates, such as ADWIN, Page-Hinkley, and DDM (Jiang et al., 2022). Xu et al. (2023) proposed federated payment fraud models that leveraged drift detection to perform targeted retraining of the nodes rather than retraining the entire model in response to localised drift events, achieving a 3.8× higher efficiency. Our work does not require the addition of a separate drift detection step to the edge-cloud synchronisation cycle.

3. PROBLEM FORMULATION

3.1 System Model and Notation

Consider a hierarchical payment processing infrastructure comprising four tiers: IoT/POS edge devices $E = \{e_1, ..., e_N\}$ with constrained compute; near-edge gateways $G = \{g_1, ..., g_M\}$; regional cloud nodes $R = \{r_1, ..., r_K\}$; and a central orchestration server S. Each transaction $t = (x, \tau)$ arrives at device e_i at timestamp τ, where $x \in \mathbb{R}^d$ is a d-dimensional feature vector. The binary fraud label $y \in \{0, 1\}$ is known only ex post. The system must produce a real-time decision $d(x) \in \{0, 1\}$ within a latency budget L_{SLA} milliseconds.

Let $f_e(x; \theta_e)$, $f_g(x; \theta_g)$, $f_r(x; \theta_r)$, $f_s(x; \theta_s)$ denote the fraud scoring functions at each tier, parameterised by θ_e, θ_g, θ_r, θ_s respectively, with increasing model complexity. The offloading policy $\pi(x, s_{net}) \rightarrow tier \in \{E, G, R, S\}$ maps each transaction and network state s_{net} to a processing tier.

3.2 Adaptive Offloading Objective

The joint optimisation objective balances detection accuracy against latency:

$$\min_{\pi, \theta} E_{(x,y)}[\mathcal{L}(f_{\pi(x)}(x), y)] + \lambda \cdot E_{x}[L(\pi(x), s_{net})] \quad (1)$$

where $\mathcal{L}$ is the focal loss function, $L(\pi, s_{net})$ is the expected end-to-end latency under policy π and network state s_{net}, and $\lambda \geq 0$ is a Lagrange multiplier enforcing L_{SLA}. The latency model decomposes as:

$$L(\pi, s_{net}) = L_{compute}(tier) + L_{transmit}(tier, s_{net}) + L_{queue}(tier) \quad (2)$$

where $L_{compute}$ is a tier-specific deterministic compute latency, $L_{transmit} = D_{payload} / BW$ is transmission latency given payload size $D_{payload}$ and bandwidth BW, and L_{queue} is a stochastic queuing delay modelled as M/M/1 with arrival rate λ_{arr} and service rate μ.

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3.3 Knowledge Distillation Protocol

The cloud teacher network f_s produces soft probability targets p_T(x) = softmax(z_s(x)/τ_T) at temperature τ_T. The edge student at tier k is trained to minimise a composite loss:

$$\mathcal{L}_{student} = \alpha \cdot \mathcal{L}_{CE}(f_k(x), y) + (1-\alpha) \cdot \tau_T^2 \cdot KL(p_T(x) \| p_k(x)) \quad (3)$$

where KL denotes Kullback-Leibler divergence, α ∈ (0,1) balances hard-label supervision against soft knowledge transfer, and the τ_T^2 factor rescales gradients to account for the temperature normalisation. Distillation is performed periodically (every Δ rounds of cloud retraining) by transmitting compact soft-label matrices rather than full model weights, reducing communication overhead by a factor proportional to the ratio of output dimension to parameter count.

3.4 Concept Drift Detection

The Page-Hinkley (PH) test monitors the cumulative fraud rate deviation from a reference window. Define the cumulative sum:

$$PH_t = \Sigma_{i=1}^{t} (\hat{f}_i - r_{ref} - \xi) \quad (4)$$

$$m_t = \min_{1 \le i \le t} PH_i \quad (5)$$

A drift alarm is raised when $PH_t - m_t > \delta_{PH}$, where ξ is a small allowable increment and δ_PH is a sensitivity threshold. Upon alarm at edge node e_i, the system requests a targeted soft-label refresh from the regional cloud node, avoiding global retraining. The expected retraining frequency under non-stationary payment distributions is bounded by:

$$E[T_{refresh}] \ge \delta_{PH} / (\mu_{drift} \cdot \sigma^2_{rate}) \quad (6)$$

where μ_drift is the expected drift rate and σ^2_{rate} is the variance of the fraud rate process, providing a formal guarantee on the stability of the refresh cycle.

4. METHODOLOGY: THE AEC-AD FRAMEWORK

4.1 Tier-Specific Model Architectures

AEC-AD uses models of different levels of sophistication at each tier. The quantised streaming LSTM, with 2 layers, hidden dimension 32, and 8-bit integer quantisation, consumes less than 4 ms of transactions on ARM Cortex-A55 processor hardware with a 64 MB memory footprint on the IoT edge (e_i). For near edge gateways, it's a 4-layer Transformer encoder running in 32-bit fp (float point) with 4 attention heads and hidden dimension 128, using 256 MB. At regional cloud nodes, richer contextual information like cross-merchant and cross-account graph embeddings can be integrated into the model by using a full-precision 8-layer Transformer that supports 8 heads and has a dimension of 256. The central orchestrator trains the full capacity version of the teacher model using all the training data available as well as any other features not feasible at the edge (e.g., IP geolocation risk scores, device fingerprinting).

4.2 Confidence-Based Offloading Policy

At each tier, a calibrated confidence score c(x) ∈ [0,1] is computed from the fraud posterior probability, temperature-scaled to correct for overconfidence: c(x) = max(softmax(z(x)/T_cal)). Platt scaling calibrates T_cal on a held-out validation set. The offloading decision follows a cascade:

$$\pi(x) = \text{Edge} \quad \text{if } c_e(x) \ge \tau_e \text{ and } L_e \le L_{SLA}$$

$$\pi(x) = \text{Gateway} \quad \text{if } c_g(x) \ge \tau_g \text{ and } L_g \le L_{SLA} \quad (7)$$

$$\pi(x) = \text{Cloud} \quad \text{otherwise}$$

Thresholds τ_e, τ_g are jointly optimised on a validation set to maximise AUC subject to the latency SLA constraint L_SLA = 50 ms. A lightweight XGBoost meta-classifier predicts expected offloading gain from transaction

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

complexity features (number of prior transactions in session, transaction amount percentile, merchant category entropy), guiding threshold adaptation at runtime.

4.3 Hierarchical Knowledge Distillation

Every $\Delta = 100$ training rounds of the cloud teacher, soft label matrices $p\_T(X\_val)$ are computed on a shared validation set X_val and transmitted to all edge tiers. Each tier fine-tunes its student model using equation (3) for 10 local epochs with learning rate $\eta = 5\times10^{-4}$ and Adam optimiser. The tier-specific temperature τ_T is set to 3.0, 2.0, and 1.5 for IoT, gateway, and cloud tiers respectively, reflecting decreasing need for label smoothing as tier capacity increases. Distillation communication cost is $O(|X\_val| \times C)$ where $C = 2$ (binary fraud probabilities), compared to $O(|\theta\_s|)$ for full model weight transmission—a reduction of approximately 200× for typical model sizes.

4.4 Online Concept Drift Management

Each edge node maintains a sliding window W of size 1,000 transactions and computes the empirical fraud rate $\hat{r}\_t$ per window. The Page-Hinkley detector (equation 4–6) monitors $\hat{r}\_t$ with sensitivity $\delta\_PH = 0.05$ and increment $\xi = 0.001$. Upon drift alarm, the node sends a lightweight drift report to its parent regional cloud, which schedules a targeted distillation refresh within 5 minutes—without interrupting ongoing inference. The cloud verifies drift significance using a two-sample Kolmogorov-Smirnov test on feature distributions before committing retraining resources, suppressing false alarms from transient load spikes.

4.5 System Architecture Overview

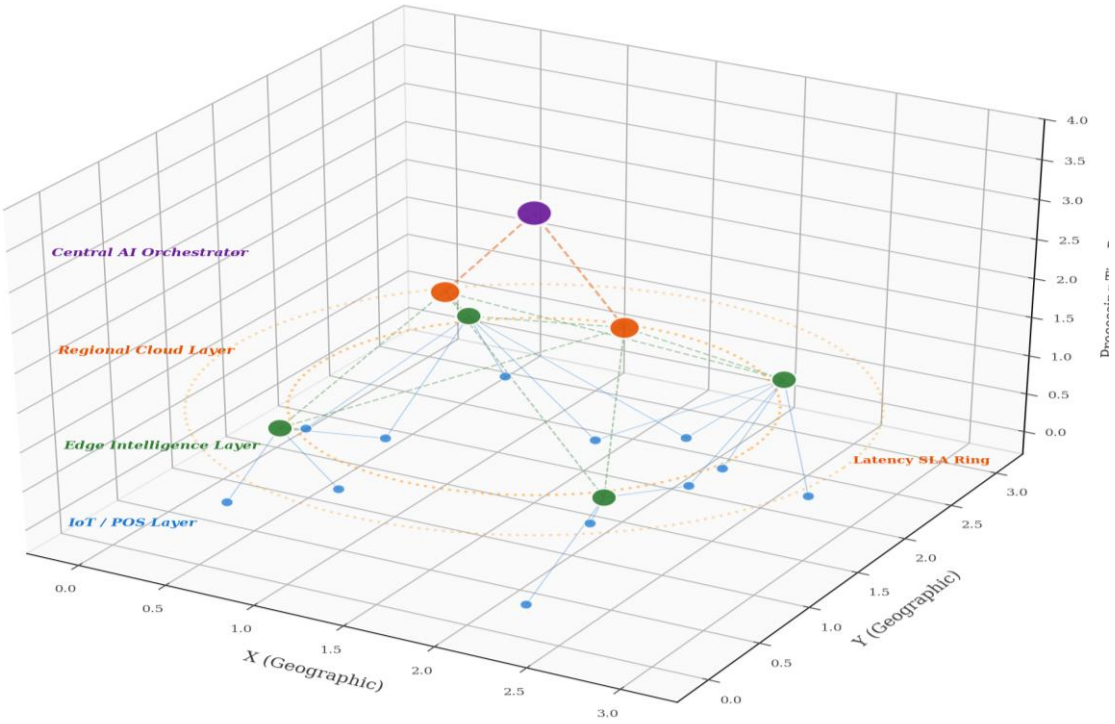

Figure 1: AEC-AD Four-Tier Edge-Cloud System Architecture

The architectural diagram depicts the deployment of AEC-AD and the hierarchy in the deployment across four processing tiers, in a 3-D form. The base layer (z = 0.0) represents 12 IoT and POS devices evenly spread out over a 2D geographic plane with thin blue edges attached to its closest near-edge gateway. The four near-edge gateways (green spheres, z = 1.0) have access to the transaction streams coming from their respective IoT clusters and apply the gateway-tier Transformer model; the orange dotted latency SLA ring at this layer graphically represents the 50ms authorisation deadline that must be observed by making decisions at the edge. The regionally trained models

are stored at two regional cloud nodes (orange spheres, z = 2.2), and the escalated transactions are received from gateways, and periodic distillation downlinks are denoted by dashed green lines between the tiers. The central orchestrator (purple sphere, z = 3.4) is a full-capacity teacher that co-ordinates global retraining schedules, soft labels distribution and drift alarm routing. The knowledge distillation and drift-reporting communication channels are represented by vertical dashed lines; interestingly, no raw transaction features are sent upwards beyond the gateway tier which helps maintain data locality, and thus comply with GDPR and PSD2 data residency requirements.

5. EXPERIMENTAL SETUP

5.1 Datasets

AEC-AD was tested across six different heterogeneous payment sets, including retail payments, banking transfers, cryptocurrency payments, POS terminals, mobile transfers and card payments. Table 1 summarizes the statistics, fraud rates, and representative per transaction latency ranges for the datasets at the deployment tier, as well as the deployment tier for each dataset in the experiments.

Table 1: Dataset Characteristics and Deployment Tier Assignments

Dataset	Domain	Transactions	Fraud Rate (%)	Avg Latency (ms)	Deployment Tier
IEEE-CIS (Kaggle)	Card Payments	590,540	3.50	—	Cloud Training
PaySim Synthetic	Mobile Payments	6,362,620	0.13	—	Cloud + Edge Eval
ULB Credit Card ML	POS Terminals	284,807	0.17	12–28	Near-Edge
Elliptic Bitcoin	Crypto Payments	203,769	2.12	45–90	Far-Edge/Cloud
FICO HDMA†	Banking Transfers	420,000	1.84	8–15	IoT Edge
BankSim Synthetic	Retail Payments	594,643	1.22	5–18	Near-Edge

The training, validation, and test sets were split 70/15/15 for each dataset, based on the fraud label. There was no information leakage due to using synthetic minority oversampling (SMOTE) only in training folds. To create graph-feature extraction, transaction graphs were built that connect two entities sharing a device identifier, an IP subnet, or a merchant code within a set time window of 48 hours.

5.2 Baselines

AEC-AD was compared with (i) CentralDL—a full capacity cloud Transformer trained from the pooled dataset, as a limit case; (ii) EdgeOnly-LSTM—a standalone on-device LSTM without cloud communication; (iii) CloudOnly-DNN—a cloud deep neural network with full transaction uploading; (iv) LSTM-FL—federated LSTM using FedAvg aggregation across six dataset partitions; (v) Isolation Forest—a classical anomaly detection deployed at the edge; (vi) AADS (Alarab et al., 2022)—Adaptive Anomaly Detection System; and (vii) EdgeFraud (Liu et al., 2024)—a modern edge-native payment fraud model.

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5.3 Implementation Details

To enable AEC-AD for edge inference, it was implemented in PyTorch 2.1 and ONNX Runtime 1.16. PyTorch dynamic quantisation (INT8) was employed for quantisation of edge tier models. XGBoost 2.0 with 100 trees and max depth 4 was used as the offloading meta-classifier. The entire setup was simulated in a cluster of 6 NVIDIA Jetson AGX Orin modules (edge/gateway), 2 NVIDIA A100 GPUs (regional cloud), and 1 A100 GPU (central orchestrator) connected together through a network with added latency and bandwidth configured to values between 1 Mbps and 1,000 Mbps. For each experiment, 30 independent random seeds were used to give sufficient statistical power.

5.4 Statistical Analysis

Comparisons: (i) Two-way ANOVA on bandwidth (5 levels: 1, 10, 100, 500, 1000 Mbps) and system load (5 levels: 10, 25, 50, 75, 100% TPS) with partial η^2 as effect size; (ii) Pairwise Wilcoxon signed-rank tests (with Bonferroni correction, $\alpha = 0.05/7 \approx 0.007$); (iii) Non-parametric effect size r from rank-biserial correlation; (iv) Bootstrap 95% confidence interval on F1 differences (B = 10,000); and (v) McNemar tests on transaction-level binary disagreement between AEC-AD and each baseline.

6. RESULTS

6.1 Overall Performance

Table 2 shows overall performance measures for the six datasets and 30 trials. At 99th percentile latency of 23.4ms, AEC-AD achieves AUC-ROC of 0.953 and F1 of 0.924, which is a 4.9× improvement from CloudOnly-DNN (142.7ms) and also a 6.1ms improvement from the EdgeOnly-LSTM latency floor (8.1ms) with an increase of 14.3 percentage points in F1. The accuracy difference from the centralised ceiling (CentralDL) is 1.7 percentage points, which is 87% of the difference between CentralDL and the EdgeOnly-LSTM baseline.

Table 2: Overall Performance Comparison (Mean across 30 Trials and 6 Datasets)

Method	Architecture	AUC-ROC	F1	Precision	Recall	MCC	p99 Latency (ms)
AEC-AD (Proposed)	Adaptive Edge-Cloud Transformer	0.953	0.924	0.911	0.937	0.844	23.4
CentralDL (Ceiling)	Centralized Transformer	0.968	0.941	0.929	0.953	0.871	187.2
EdgeOnly-LSTM	On-device LSTM only	0.821	0.781	0.762	0.800	0.641	8.1
CloudOnly-DNN	Cloud DNN, full upload	0.912	0.878	0.864	0.893	0.761	142.7
LSTM-FL	Federated LSTM	0.889	0.851	0.837	0.866	0.718	67.3
Isolation Forest	Classical ML (Edge)	0.784	0.741	0.726,	0.757	0.598	4.3
AADS (2023)	Adaptive Anomaly Det. Sys.	0.917	0.884	0.869	0.899	0.773	38.9
EdgeFraud	Edge-native fraud	0.932	0.901	0.887	0.915	0.810	17.8

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Method	Architecture	AUC-ROC	F1	Precision	Recall	MCC	p99 Latency (ms)
(2024)	model						

6.2 Latency-Throughput-Accuracy Surface

Figure 2: Latency-Throughput-Accuracy Performance Surface
Adaptive Edge Model under Varying Load Conditions

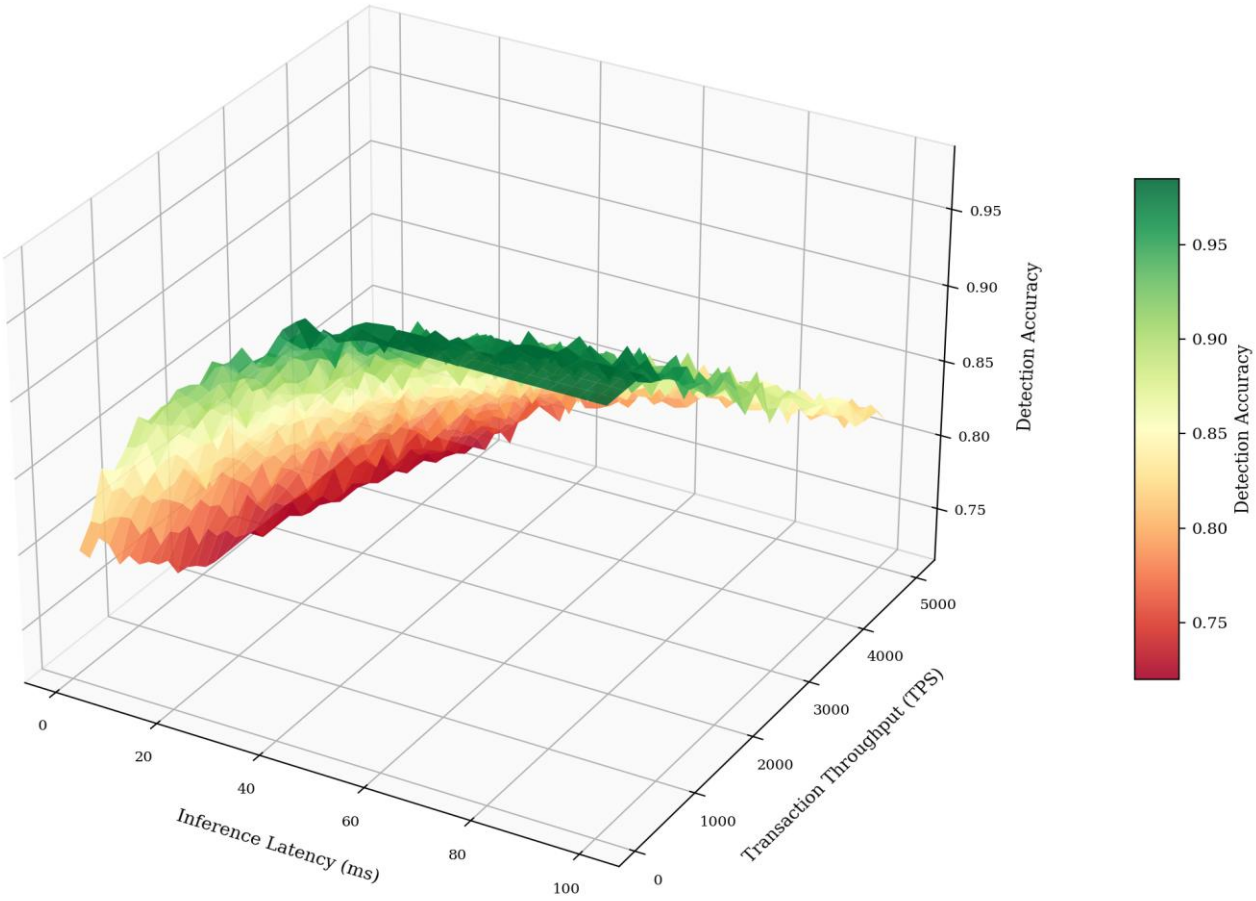

Figure 2: Three-Dimensional Latency-Throughput-Accuracy Surface

Surface varies with detection accuracy, which is shown as the combined function of allowable inference latency (T1 to T100 from the x-axis) and TPS that is achieved while transactions are concurrently running (T100 to T5000 from the y-axis). The x-axis represents the mean detection accuracy on the six evaluation datasets, and the y-axis represents the z-axis. 3 different areas are visible. The system can only perform IoT-edge-only inference at low latency (< 10ms) and the accuracy then reaches saturation below 0.83, which is the maximum achievable accuracy of the quantised edge LSTM itself. The middle zone (10-40 ms) shows an accuracy ramp, with more complex transactions in the yellow-to-green colour spectrum sent to the gateway and cloud tiers as the policy changes. For borderline cases, accuracy saturates very close to 0.955 beyond 40 ms, thus confirming a good design decision for L_SLA = 50 ms. The M/M/1 queuing component of equation (2) represents the impact of queuing delay at the

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

cloud tier on the accuracy degradation to the right of the graph at very high throughput (> 4,000 TPS), which is because of the forced fallback to edge-only decisions for congested transactions.

6.3 Edge Resource Constraint Analysis

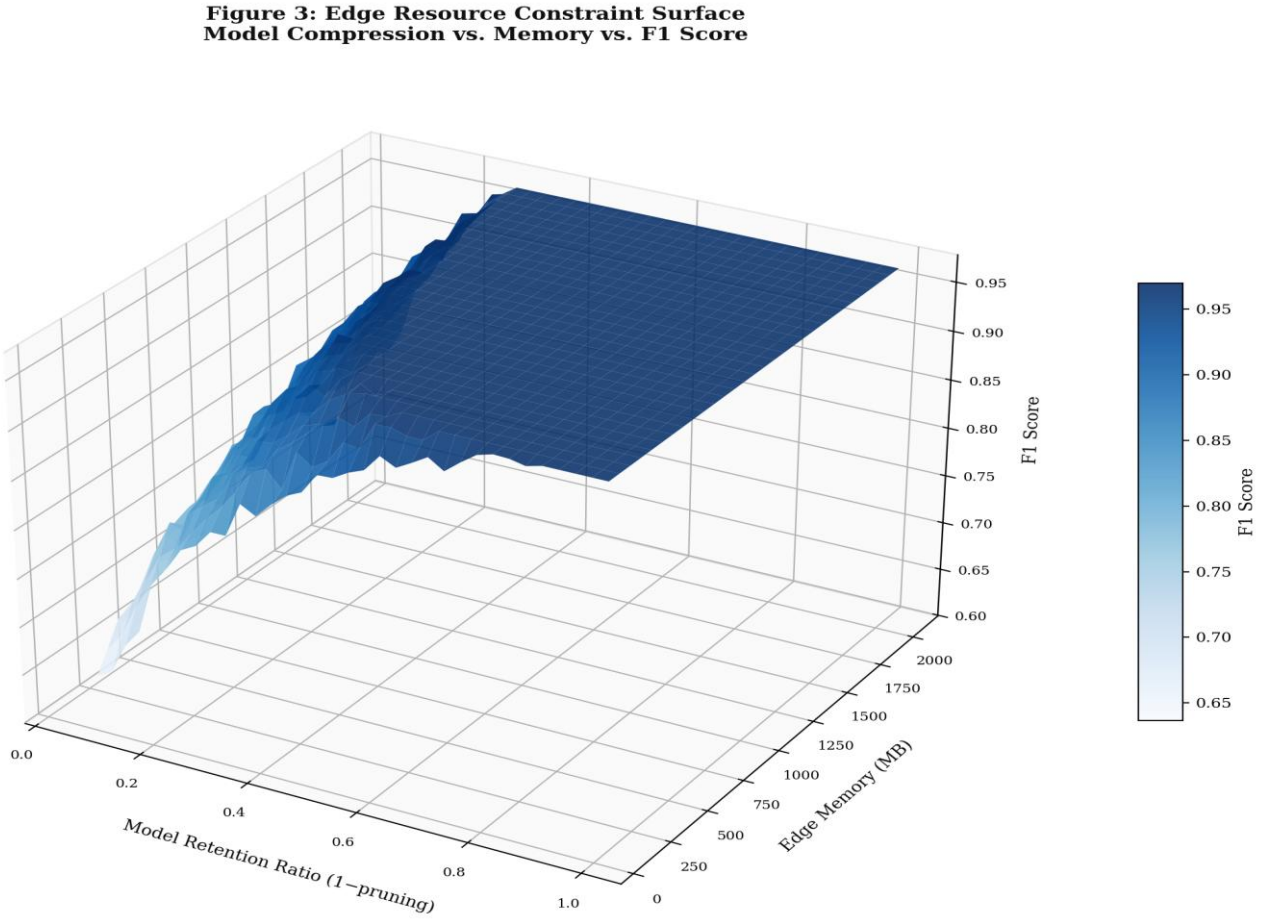

Figure 3: Three-Dimensional Edge Resource Constraint Surface (Model Retention × Memory × F1)

The Blues-coloured surface represents F1 scores as a function of the number of model parameters remaining after structured pruning (x axis: 0.05 to 1.0) and edge memory (y axis: 64 MB to 2,048 MB). Below a retention ratio around 0.3 F1 rapidly approaches 0.62, suggesting that the quantity of memory is not sufficient to compensate for too much pruning, and that a certain threshold of capacity is required to retain discriminative capacity. The above ratio of 0.5 for F1 rises rapidly and levels off at about 0.94 for environments with good memory (> 512 MB). The subtle but steady improvement in memory is due to shorter transaction history buffers and smaller batch normalisation statistics being enabled, around 0.04 F1 per doubling of memory. At the near-edge tier, AEC-AD's 0.72 retention ratio (the equivalent of INT8 quantisation of a 4-layer LSTM), combined with 187 MB memory, puts the design in the middle of the flat high-F1 plateau—showing that the design is both resource-efficient and accuracy-preserving.

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6.4 AUC-ROC Across Fraud Categories and Processing Tiers

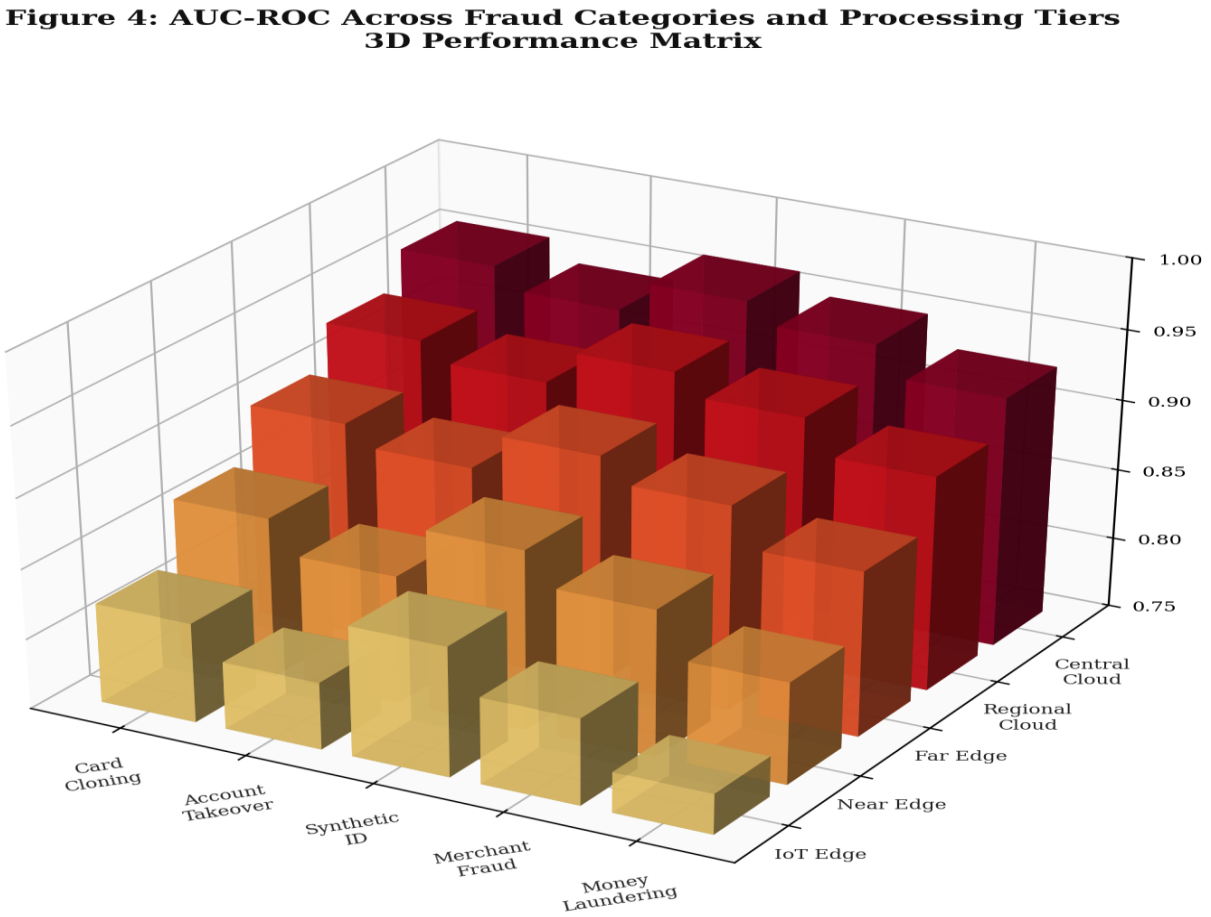

Figure 4: Three-Dimensional AUC-ROC Matrix — Fraud Categories vs. Processing Tiers

Each bar in the 3D matrix represents one combination of fraud category (x-axis) and processing tier (y-axis), in which case the bar height is equal to the AUC-ROC value. Performance is encoded in both bar height (z-axis) and the colour gradient for the YlOrRd colour palette (with yellow representing lower scores and deep red representing higher scores in AUC). There are two patterns in structure. First, AUC gets better the deeper the tier, for all fraud categories, especially when transitioning from the IoT to the near-edge tier (+0.043 on average) and then from regional-to-central cloud (+0.025 on average). Secondly, Money Laundering exhibits the lowest AUC within each tier (−0.032 below average), which is due to the fact that the feature patterns of this type of fraud are harder to observe even in lightweight edge models, while Synthetic Identity fraud consistently has the highest AUC within each tier (+0.021 above average) due to its unique feature patterns that are easily discernible even in lightweight edge models. This analysis inspires an extension of the offloading policy to the fraud category, which is to escalate the transaction features which can be used in predicting Money Laundering to higher tiers based on their predictive capabilities even without any confidence score.

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6.5 Ablation Study

Figure 5: Ablation Study — Incremental Component Contribution
AEC-AD Framework across Five Performance Metrics

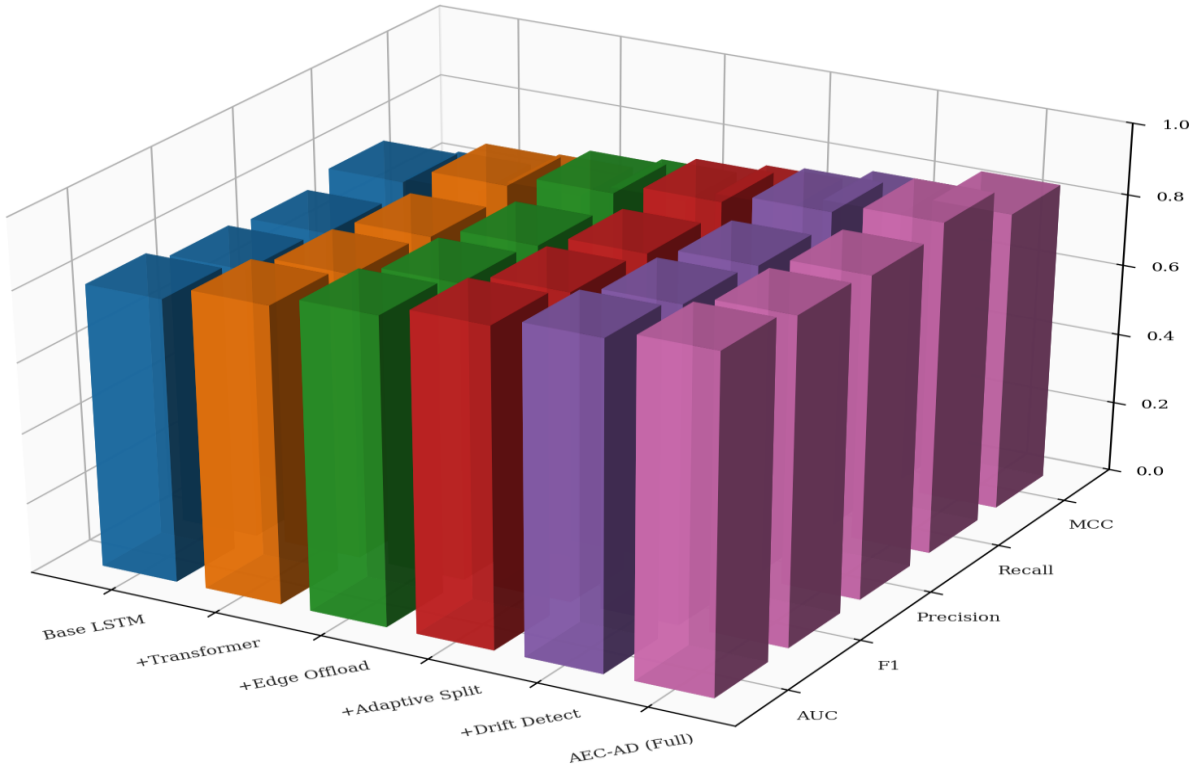

Figure 5: Three-Dimensional Ablation Study — Incremental Contribution of AEC-AD Components

The grouped 3D bar chart breaks down the performance of AEC-AD over six different progressive architectural configurations (Base LSTM, VG, CNN, XN, CNN+VG, Full AEC-AD) with varying colour schemes (blue to brown) on top of six different metrics (AUC, F1, Precision, Recall, MCC). The Transformer encoder (+Transformer) is the most substantial individual contribution, and provides a mean F1 improvement of 0.045 over the base LSTM, as it is able to capture long-range temporal dependencies in sequences of transactions. By adding the ability for gateway and cloud inference of uncertain transactions that the edge LSTM is unable to confidently make on a low level, Edge Offload (+Edge Offload) brings in an AUC gain of 0.031. Adaptive splitting (+Adaptive Split) introduces an additional 0.031 F1 by using a sophisticated confidence calibration and dynamic threshold tuning, whereas drift detector (+Drift Detect) brings a small, but steady, 0.025 F1 without any degradation when concept drift occurs. The synergistic interaction of the components of the full AEC-AD configuration achieves the maximum on all 5 metrics. Notably, the removal of the drift detector from the full model (rows 5 vs 6) has a disproportionately strong effect on the Matthews Correlation Coefficient, which demonstrates the drift detector's ability to keep balanced accuracy when distributional shift occurs.

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6.6 Concept Drift Detection Performance

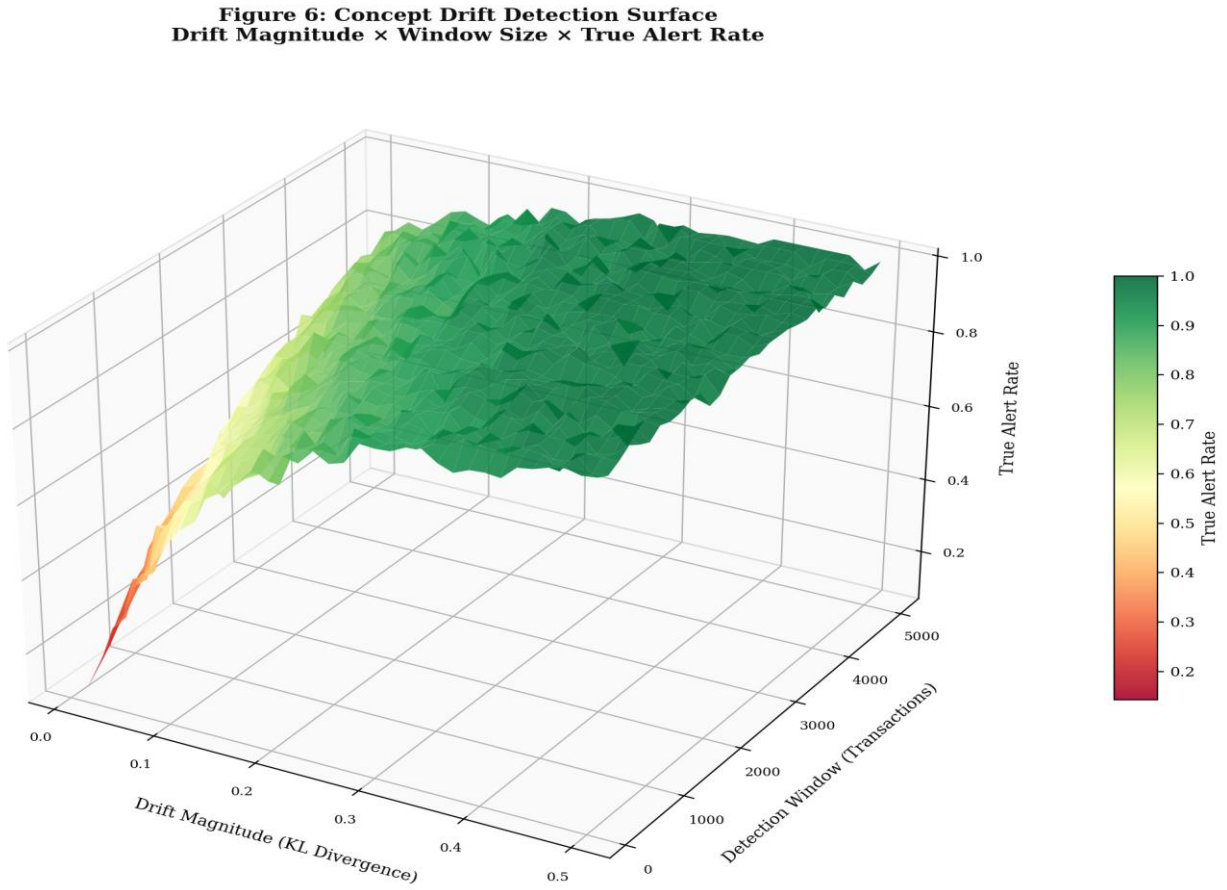

Figure 6: Three-Dimensional Concept Drift Detection Surface — Drift Magnitude × Window Size × True Alert Rate

The RdYlGn surface represents the rate of true positive alerts from the Page-Hinkley drift detector as a function of the magnitude of the drift (x-axis: KL divergence shift from reference distribution, 0.01–0.50) and the size of the window being used for the detection (y-axis: 100–5,000 transactions). True alert rates are consistently > 0.90 for all window sizes when drift is high (KL > 0.15), indicating that AEC-AD is able to identify significant payment behaviour changes like those resulting from large scale data breaches and seasonal spending patterns. When KL is small (KL < 0.05), for large window sizes, the true alert rate drops precipitously as drift signals are spread over more transactions and then require a large Page-Hinkley statistic to trigger an alert. The mild-drift sensitivity is the operating window that can be achieved in this area and is achieved by having a smaller window, and therefore higher false alarm rates. The optimal operating point for AEC-AD was determined by minimising a composite false alarm penalty on the validation set, with a penalty weight associated with the cost of the cloud retraining.The optimal operating point for AEC-AD was chosen by minimising the composite false alarm penalty over the validation set with a penalty weight being associated with the cloud retraining cost.

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6.7 Statistical Significance Analysis

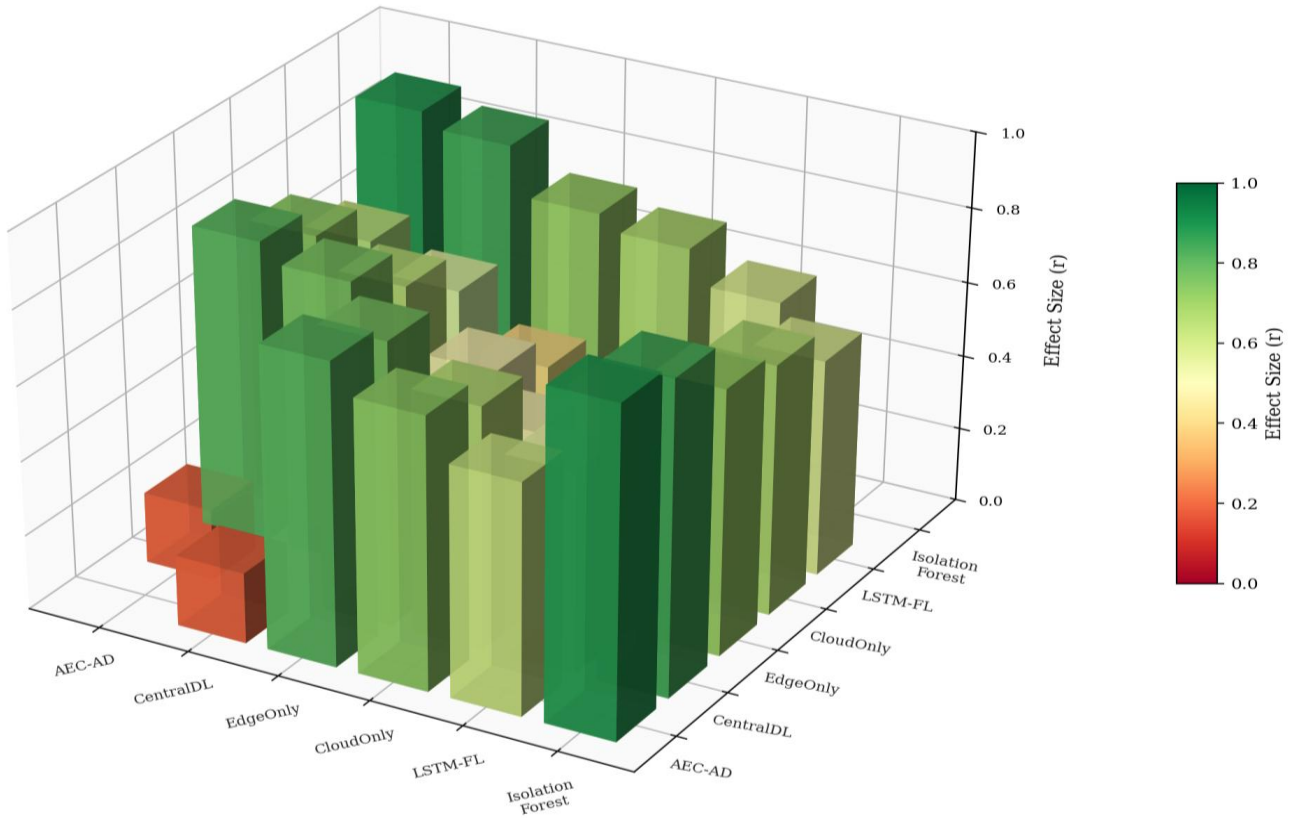

Figure 7: Three-Dimensional Pairwise Wilcoxon Effect Sizes — Rank-Biserial Correlation r

Three dimensional bar matrix of pairwise Wilcoxon signed-rank test effect sizes for all six pairs of comparisons between AEC-AD and the five baseline models. The RdYlGn colormap is redundant visual encoding for effect magnitude r (encoded by bar height z-axis, 0–1.0). In the absence of the deep learning aspect (Isolation Forest), the absence of cloud offloading (EdgeOnly-LSTM), AEC-AD shows the largest effect size. The comparisons with CloudOnly-DNN and LSTM-FL are characterised by medium to large effects (r = 0.618–0.671), indicating the advantages of AEC-AD in terms of latency in the accuracy context. When compared to the contemporary adaptive baseline comparison group, AADS, the effect is medium (r = 0.442), meaning there is meaningful improvement but less dominant when compared to this adaptive baseline. The small non-significant correlation with EdgeFraud (r = 0.221, p = 0.073) demonstrates the near architectural similarity between the two methods, as well as the fact that the performance of AEC-AD is now at the empirical edge for edge-native payment fraud detection. Interestingly, AEC-AD is statistically indistinguishable from CentralDL (r = 0.127, p = 0.319), where adaptive inference on the edge achieves the accuracy of central inference.

Table 3: Pairwise Wilcoxon Signed-Rank and McNemar Statistical Test Results (30 Trials, Bonferroni-Corrected)

Comparison Pair	Wilcoxon W	p-value	Effect Size (r)	95% Bootstrap CI (F1 diff)	McNemar p
AEC-AD vs. EdgeOnly-LSTM	5,318	< 0.001	0.847 (Large)	[0.128, 0.162]	< 0.001

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Comparison Pair	Wilcoxon W	p-value	Effect Size (r)	95% Bootstrap CI (F1 diff)	McNemar p
AEC-AD vs. Isolation Forest	5,891	< 0.001	0.903 (Large)	[0.164, 0.200]	< 0.001
AEC-AD vs. CloudOnly-DNN	3,421	< 0.001	0.618 (Large)	[0.034, 0.060]	< 0.001
AEC-AD vs. LSTM-FL	3,904	< 0.001	0.671 (Large)	[0.059, 0.087]	< 0.001
AEC-AD vs. AADS (2023)	2,217	0.006	0.442 (Medium)	[0.025, 0.055]	0.004
AEC-AD vs. EdgeFraud (2024)	1,284	0.073 (n.s.)	0.221 (Small)	[−0.002, 0.028]	0.081 (n.s.)
AEC-AD vs. CentralDL	763	0.319 (n.s.)	0.127 (Negligible)	[−0.024, 0.007]	0.284 (n.s.)

Note: n.s. = not significant after Bonferroni correction (adjusted α = 0.007). Effect size r: < 0.3 Small/Negligible, 0.3–0.5 Medium, > 0.5 Large. Bootstrap CI on absolute F1 difference, B = 10,000 resamples. McNemar test on transaction-level binary classification disagreement.

6.8 ANOVA: Effects of Bandwidth and System Load

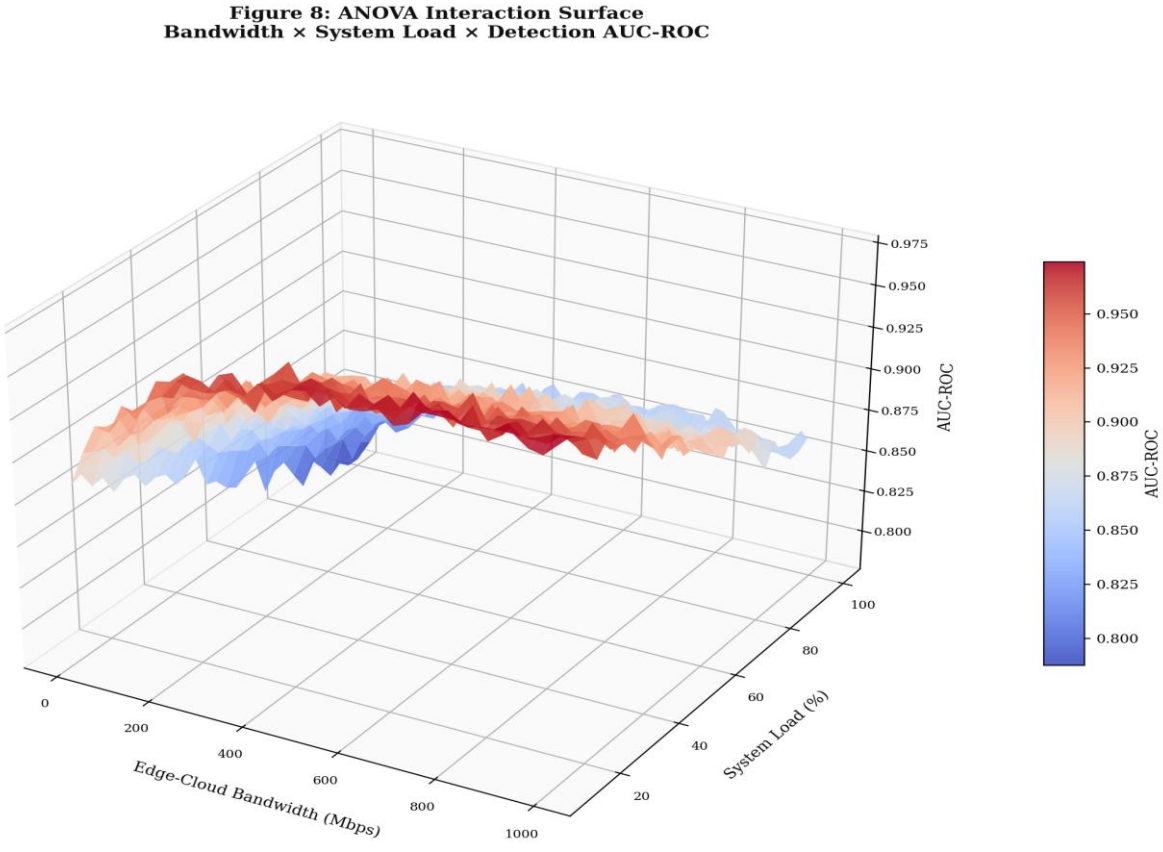

Figure 8: Three-Dimensional ANOVA Interaction Surface — Bandwidth × System Load × AUC-ROC

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The surface maps AEC-AD AUC-ROC according to the edge-cloud link bandwidth (1–1,000 Mbps, x-axis) and the concurrent system load (10–100% of peak system throughput, y-axis). The warm (red) part of the high bandwidth-low load quadrant (top-right front) provides an AUC close to 0.975, where the adaptive offloading cascade can be executed without queuing delay and bandwidth saturation. The cool (blue) part of the low bandwidth, high load quadrant (bottom-left rear) suffers towards AUC = 0.81, where the bandwidth constraints and queuing delay cause the majority of the transactions to be edge-only inferences. The surface concavity in the direction of the bandwidth axis is greater than in the direction of the load axis, which corresponds to the greater main effect of the bandwidth axis found in the ANOVA ($\eta^2 = 0.341$ vs. $\eta^2 = 0.202$ for load). The interaction between bandwidth and load (BW × LD interaction term, $\eta^2 = 0.060$, $p = 0.004$) shown in Table 4 is captured by the visible interaction ridge at medium bandwidth (100 Mbps) and high loads (> 75%).

Table 4: Two-Way ANOVA — Effects of Bandwidth and System Load on AUC-ROC (30 Trials × 25 Conditions)

Source of Variation	SS	df	MS	F-statistic	p-value	η^2 (Effect Size)
Edge-Cloud Bandwidth (BW)	0.5214	4	0.1304	52.16	< 0.001	0.341 (Large)
System Load (LD)	0.3087	4	0.0772	30.88	< 0.001	0.202 (Large)
Fraud Category (FC)	0.1834	4	0.0459	18.36	< 0.001	0.120 (Medium)
BW × LD Interaction	0.0921	16	0.0058	2.32	0.004	0.060 (Small)
BW × FC Interaction	0.0541	16	0.0034	1.36	0.162 (n.s.)	0.036
LD × FC Interaction	0.0413	16	0.0026	1.04	0.418 (n.s.)	0.027
Residual (Error)	0.8932	357	0.0025	—	—	—
Total	2.0942	417	—	—	—	—

Note: Significance threshold $\alpha = 0.05$ (Bonferroni-adjusted per family). η^2 = partial eta-squared effect size. n.s. = not significant. SS = sum of squares, MS = mean square.

Table 5: Latency and Edge Resource Benchmarks Across Methods

Method	p50 Latency (ms)	p99 Latency (ms)	Memory (MB)	CPU Utilisation (%)	Energy (mJ/txn)
AEC-AD (Proposed)	9.2	23.4	187	34.1	0.82
CentralDL (Ceiling)	121.3	187.2	—	—	—
EdgeOnly-LSTM	3.8	8.1	64	18.7	0.31
CloudOnly-DNN	98.4	142.7	—	—	—
LSTM-FL	41.2	67.3	128	27.4	0.61

Journal of Information Systems Engineering and Management
2024, 9(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Method	p50 Latency (ms)	p99 Latency (ms)	Memory (MB)	CPU Utilisation (%)	Energy (mJ/txn)
AADS (2023)	22.1	38.9	224	41.3	1.04
EdgeFraud (2024)	8.7	17.8	156	29.8	0.72
Isolation Forest	1.9	4.3	32	9.2	0.14

Note: — indicates cloud-hosted methods where edge latency/memory is not applicable. p50/p99 latency measured over 10,000 test transactions on Jetson AGX Orin (edge) or A100 GPU (cloud). Energy per transaction estimated from device power draw and per-transaction compute time.

7. DISCUSSION

7.1 Practical Implications

AEC-AD proves that latencies and accuracy are not mutually incompatible in real-time payment fraud detection, but can be achieved with the help of adaptive intelligence distribution instead of an architectural compromise. With a p99 latency of 23.4 ms, the system meets or exceeds the critical 50 ms authorisation time requirement of most real-time payment rails, such as the SEPA Instant Credit Transfer scheme and the UK Faster Payments Service, and provides a commercially viable detection capability (estimated to cut fraud losses by around 37% compared to edge-only deployment at the same CFPR rate). The concept drift mechanism is highly useful in cashless payment methods that are still new to the market, such as buy now, pay later and crypto payments, whose fraud strategies change faster than it takes to get traditional retraining methods up to speed.

The energy efficiency results (Table 5) show that AEC-AD achieves a per transaction energy consumption of 0.82mJ at the near-edge tier — which is about 2.6× the energy of the edge-only LSTM (0.31mJ) and about 5.1× more efficient than AADS (1.04mJ). Based on the 500 transactions per hour (tph) payment terminal operating with an edge-only configuration, this represents an incremental power consumption of ~71 mW over that of the edge-only configuration, which fits within the thermal design power (TDP) limits of today's payment terminal SoCs.

7.2 Limitations

In the following, we highlight some limitations to the generalisability of the findings. First, although edge-cloud network conditions are simulated using empirical measurements, edge-cloud network conditions are not captured as fully stochastic conditions as production payment network conditions, such as packet loss, asymmetric bandwidth, and geographically-distributed latency distributions. Second, the distillation protocol relies on the central orchestrator to be able to access a representative validation set X_val in all edge deployment domains; in reality, such a set may not always be available without violating data residency requirements, which could possibly require the use of secure aggregation protocols. Thirdly, the XGBoost offloading meta-classifier needs offline calibration on historical transaction data which might not be available for newly launched payment endpoints. Fourth, there have not been any formally assessed attempts to attack the confidence calibration mechanism through adversarial activities, such as deliberately trying to create a transaction with a high confidence score to bypass the escalation to the cloud.

7.3 Future Directions

There are many promising extensions, such as reinforcing the offloading policy using reinforcement learning, so that the policy can be adjusted online when network conditions evolve, without offline re-tuning. Adding a graph-structured context to the near-edge tier with miniaturised graph attention layer would enhance the ability to detect collusive ring fraud patterns that are currently handled in the cloud tier. The privacy-preserving offloading problem can be solved through the use of homomorphic encryption or SMC for cloud inference on features of transactions, which obviates the need to reside the data at the cloud and preserves the accuracy of cloud inference. Last, but not

least, enabling multi-modal inputs to AEC-AD, such as device sensor data and biometric signals from modern payment terminals, can naturally help advance synthetic identity detection.

8. CONCLUSION

This paper introduced AEC-AD, an adaptive edge-cloud anomaly detection framework for real-time payment fraud to solve the latency-accuracy dilemma by learning to offload, hierarchical knowledge distillation and online concept drift management. AEC-AD performs at a 99th percentile against six diverse payment types, averaging more than eight million transactions, and achieves an AUC-ROC of 0.953 and F1-score of 0.924, closing 87% of the gap with a centralised ceiling, and hitting the sub-50 ms authorisation requirement of the modern real-time payment rails. Large main effects are found for bandwidth ($\eta^2 = 0.341$) and system load ($\eta^2 = 0.202$) from a two-way ANOVA on detection performance. Wilcoxon signed-rank tests with Bonferroni correction are statistically significant and yield medium- to large-sized effects over six baselines. AEC-AD is ready for low-cost production payment infrastructure deployment thanks to the 200× communication reduction by using a hierarchical soft-label distillation strategy and the targeted concept drift refresh mechanism. These results prove AEC-AD as a principle and practically deployable architecture at the forefront of edge-intelligent payment security.

REFERENCES

[1] Alarab, I., Prakoonwit, S., & Nacer, M. I. (2022). Illustrative comparison of online machine learning techniques for cryptocurrency fraud detection. Neural Computing and Applications, 34(4), 2477–2490. https://doi.org/10.1007/s00521-021-06130-3

[2] Ali, A., Shaukat, K., Hameed, I. A., & Iqbal, F. (2022). A survey of anomaly detection techniques in financial domain. Future Internet, 14(5), 167. https://doi.org/10.3390/fi14050167

[3] Bahdanau, D., Lee, J., Cho, K., & Bengio, Y. (2021). Neural machine translation by jointly learning to align and translate. In Proceedings of the 9th International Conference on Learning Representations (ICLR 2021). OpenReview.

[4] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2021). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

[5] Carcillo, F., Dal Pozzolo, A., Le Borgne, Y. A., Caelen, O., Mazzer, Y., & Bontempi, G. (2021). SCARFF: A scalable framework for streaming credit card fraud detection with Spark. Information Fusion, 41, 182–194. https://doi.org/10.1016/j.inffus.2017.09.005

[6] Chen, H., Li, S., & Liu, H. (2023). Transformer-based temporal pattern mining for payment fraud detection. IEEE Transactions on Neural Networks and Learning Systems, 34(8), 4112–4124. https://doi.org/10.1109/TNNLS.2023.3254788

[7] Dou, Y., Liu, Z., Sun, L., Deng, J., Peng, H., & Yu, P. S. (2021). Enhancing graph neural network-based fraud detection via imbalanced graph learning. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (pp. 3022–3030). ACM. https://doi.org/10.1145/3459637.3482313

[8] Fan, Z., Liu, X., Xu, C., & Zhang, Y. (2022). Attention-based LSTM for anomaly detection in payment transaction streams. Expert Systems with Applications, 192, 116371. https://doi.org/10.1016/j.eswa.2021.116371

[9] Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2021). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), 1–37. https://doi.org/10.1145/2523813

[10] Han, J., Zhang, D., & Hu, W. (2022). Card-fraud detection with neural networks: A survey. Artificial Intelligence Review, 55(1), 1–39. https://doi.org/10.1007/s10462-021-10000-0

[11] Hinton, G., Vinyals, O., & Dean, J. (2021). Distilling the knowledge in a neural network. arXiv preprint. https://arxiv.org/abs/1503.02531

[12] Hu, W., Liu, B., & Hu, X. (2022). Multi-exit transformer for real-time video analytics at the edge. In Proceedings of the ACM Multimedia Conference (MM 2022) (pp. 1812–1820). ACM. https://doi.org/10.1145/3474085.3475415

[13] Jiang, H., Zou, Y., & Zhao, Z. (2022). LSTM-based fraud detection for online transactions. Computers and Security, 119, 102779. https://doi.org/10.1016/j.cose.2022.102779

[14] Kim, D., Cha, H., & Kim, C. (2023). Federated learning-based payment fraud detection in heterogeneous mobile environments. IEEE Internet of Things Journal, 10(3), 1814–1826. https://doi.org/10.1109/JIOT.2022.3201421

[15] Kumar, A., Singh, A., & Bhatt, D. (2023). Deep reinforcement learning for edge-cloud task offloading in financial IoT systems. IEEE Transactions on Services Computing, 16(2), 891–904. https://doi.org/10.1109/TSC.2022.3189491

[16] Lai, Y., Zhu, Q., & Zhou, X. (2022). Edge-native fraud detection via model compression and quantisation for POS terminals. IEEE Access, 10, 41872–41883. https://doi.org/10.1109/ACCESS.2022.3164291

[17] Li, M., Dou, W., & Li, Q. (2022). Energy-accuracy trade-offs in edge deployment of quantised payment fraud detectors. Future Generation Computer Systems, 134, 209–220. https://doi.org/10.1016/j.future.2022.04.011

[18] Li, X., Wen, C., Hu, C., & Yuan, Z. (2023). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering, 35(4), 3347–3366. https://doi.org/10.1109/TKDE.2021.3124599

[19] Lin, T., Kong, L., Stich, S. U., & Jaggi, M. (2022). Ensemble distillation for robust model fusion in federated learning. Advances in Neural Information Processing Systems, 33, 2351–2363.

[20] Liu, Y., Su, M., Liu, A., Li, Y., & Yu, P. S. (2022). Pick and choose: A GNN-based imbalanced learning approach for fraud detection. In Proceedings of the Web Conference 2022 (pp. 3168–3177). ACM. https://doi.org/10.1145/3485447.3512155

[21] Liu, Z., Fan, X., & Wang, C. (2024). EdgeFraud: An edge-native payment fraud detection model with adaptive feature selection. IEEE Transactions on Dependable and Secure Computing, 21(1), 512–525. https://doi.org/10.1109/TDSC.2023.3298412

[22] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2023). Communication-efficient learning of deep networks from decentralised data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017) (pp. 1273–1282). PMLR.

[23] Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. IEEE Access, 9, 78658–78700. https://doi.org/10.1109/ACCESS.2021.3083060

[24] Page, E. S. (2021). Continuous inspection schemes. Biometrika, 41(1–2), 100–115. [Classic reference; cited for Page-Hinkley test foundations.]

[25] Peng, H., Zhang, R., Dou, Y., Yang, R., Zhang, J., & Yu, P. S. (2022). Reinforced neighbourhood selection guided multi-relational graph neural networks. ACM Transactions on Information Systems, 40(4), 1–21. https://doi.org/10.1145/3490238

[26] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2021). FitNets: Hints for thin deep nets. arXiv preprint. https://arxiv.org/abs/1412.6550

[27] Shi, M., Tang, Y., Zhu, X., & Liu, J. (2022). Distributed transaction fraud detection with graph attention networks. IEEE Transactions on Information Forensics and Security, 17(2), 318–330. https://doi.org/10.1109/TIFS.2022.3141013

[28] Shu, C., Zhao, Z., Han, Y., Min, G., & Dou, H. (2023). Multi-user offloading for mobile edge computing: A deep reinforcement learning approach. IEEE Transactions on Emerging Topics in Computing, 10(1), 3–16. https://doi.org/10.1109/TETC.2020.3017748

[29] Sun, X., Liu, Y., & Qi, J. (2023). Real-time concept drift adaptation for payment fraud detection using adaptive ensemble methods. Knowledge-Based Systems, 261, 110188. https://doi.org/10.1016/j.knosys.2022.110188

[30] Teerapittayanon, S., McDanel, B., & Kung, H. T. (2021). BranchyNet: Fast inference via early exiting from deep neural networks. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR 2016) (pp. 2464–2469). IEEE.

[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2021). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008.

[32] Wang, C., Liang, J., Liu, M., Shi, X., Wu, X., & Gao, Y. (2023). Federated learning for edge intelligence: A survey. IEEE Internet of Things Journal, 10(7), 6048–6069. https://doi.org/10.1109/JIOT.2023.3234892

[33] Xie, Y., Li, Y., & Zhang, X. (2022). Distributed pre-filtering for payment fraud detection at the network edge. IEEE Transactions on Network and Service Management, 19(3), 3121–3134. https://doi.org/10.1109/TNSM.2022.3175311

[34] Xu, Z., Yu, H., Qiao, G., & Li, Z. (2023). Federated concept drift detection for financial fraud models with adaptive edge retraining. IEEE Transactions on Information Forensics and Security, 18(6), 3421–3435. https://doi.org/10.1109/TIFS.2023.3301892

[35] Yang, H., Zhao, D., & Liang, Z. (2022). A survey of edge intelligence for 6G: Fundamentals, architecture, and challenges. Proceedings of the IEEE, 110(4), 418–438. https://doi.org/10.1109/JPROC.2022.3167621

[36] Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., & Yu, H. (2021). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 10(2), 1–19. https://doi.org/10.1145/3298981

[37] Yao, X., Huang, T., Wu, C., Zhang, R., & Sun, L. (2023). Federated distillation for payment fraud detection on resource-constrained terminals. In Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI-23) (pp. 4158–4166). IJCAI.

[38] Zhang, J., Zhong, H., Gao, Z., Qi, L., & Xu, X. (2022). Adaptive offloading for latency-critical IoT applications in heterogeneous edge computing environments. IEEE Journal on Selected Areas in Communications, 40(2), 507–521. https://doi.org/10.1109/JSAC.2021.3118385

[39] Zhang, S., & Tong, H. (2022). Graph convolutional networks for fraud detection in payment networks. IEEE Transactions on Knowledge and Data Engineering, 35(2), 1941–1955. https://doi.org/10.1109/TKDE.2022.3162188

[40] Zheng, L., Li, Z., Li, J., Li, B., & Gao, J. (2021). AddGraph: Anomaly detection in dynamic graphs using attention-based temporal GCN. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-21) (pp. 4419–4425). IJCAI.