Research Article

Anomaly Detection System for Distributed Job Processing within Microservice Architectures

Ramazan Pekin[1*], Kerem Bozkurt[2]

[1] Turkcell Ödeme ve Elektronik Para Hizmetleri A.Ş. (Paycell R&D Center),
Orcid ID: https://orcid.org/0009-0007-3294-0709,
E-mail: ramazan.pekin@turkcell.com.tr
[2] Turkcell Ödeme ve Elektronik Para Hizmetleri A.Ş. (Paycell R&D Center),
Orcid ID: https://orcid.org/0009-0006-8780-1880,
E-mail: kerem.bozkurt@turkcell.com.tr
[*] Correspondence: ramazan.pekin@turkcell.com.tr; Tel.: +90 (532)6748019

Received: 26 July 2025
Revised: 20 October 2025
2nd **Revised:** 05 December 2025
3rd **Revised:** 14 December 2025
Accepted: 17 December 2025
Published: 21 December 2025

Reference: Pekin, R., & Bozkurt, K. (2025). Anomaly detection system for distributed job processing within microservice architectures. *The European Journal of Research and Development, 5*(1), 581–598.

Abstract

Mobile payment systems process millions of transactions daily across distributed microservice architectures, where operational anomalies and silent failures can lead to financial losses and system instability. Traditional threshold-based monitoring is insufficient for detecting subtle, context-dependent deviations that evolve with user behavior and workload patterns. This study introduces a self-learning hybrid anomaly detection framework that integrates Isolation Forest, LSTM Autoencoder, and One-Class SVM to capture statistical, temporal, and structural deviations in operational metrics. Model outputs are fused using a calibrated soft majority voting strategy based on normalized anomaly scores. The trained framework is deployed as a containerized microservice, enabling real-time anomaly assessment based on live operational statistics. Experimental evaluation across a fifteen-month dataset demonstrates that the ensemble improves detection robustness and reduces false negatives compared to individual models and simple averaging strategies. The results highlight the system's ability to detect silent failures and *abnormal behaviors that occur without explicit exceptions while maintaining scalability and adaptability in complex financial microservice environments.*

Keywords: Anomaly Detection; Distributed Systems; Microservices; LSTM Autoencoder; Isolation Forest; One-Class SVM; Ensemble Learning; Soft Majority Voting; Silent Failure; Mobile Payments.

1. Introduction

The increasing adoption of mobile payment applications has driven the evolution of highly distributed microservice architectures capable of processing vast volumes of transactions in real time [1]. These architectures support scalability and modularity, yet they also introduce new operational challenges, particularly in environments where asynchronous communication, background queues, and event-driven execution play central roles. One of the most critical challenges is the emergence of silent failures, which refer to degraded or abnormal system behaviors that propagate without producing explicit error statistics or exceptions. Such failures manifest subtly through abnormal fluctuations in throughput, transaction outcomes, or latency patterns, and their impact often becomes visible only after they cascade into dependent components.

Traditional rule-based monitoring solutions built on static thresholds or fixed alert rules struggle to detect these failures due to their inability to adapt to evolving workloads, seasonal variations, and user behavior. Static rules quickly become obsolete and either trigger frequent false alarms or fail to detect deviations that fall outside predefined boundaries. To overcome these limitations, research has increasingly focused on machine learning–based anomaly detection methods capable of learning from historical patterns and identifying deviations without manual threshold tuning [2],[3],[4]. However, deploying such models in production-grade, distributed financial environments requires careful consideration of scalability, interpretability, and integration overhead.

This study proposes a hybrid, self-learning anomaly detection system that synthesizes complementary unsupervised learning methods to identify abnormal behavioral patterns in financial microservice execution traces. The system leverages Isolation Forest for statistical outlier detection, an LSTM Autoencoder for modeling temporal dependencies, and One-Class SVM for boundary-based deviation detection. The ensemble's soft-voting fusion mechanism ensures balanced sensitivity across different anomaly types, while an adaptive thresholding method based on conformal quantiles provides robustness under shifting operational conditions. The contributions of this work include:

- a hybrid ensemble detection framework optimized for microservice telemetry

- a self-learning mechanism enabling autonomous adaptation under drift
- a containerized real-time inference service
- an empirical evaluation on long-term production-like operational data

The system is containerized to ensure scalability and seamless deployment within modern cloud-native financial infrastructures.

2. Materials and Methods

2.1 System Architecture

The proposed system is composed of four interconnected layers. The first layer is the data collection component, which gathers operational statistics from the distributed microservice infrastructure, capturing metrics such as success and failure counts as well as latency information. The second layer is the feature engineering pipeline, which transforms unstructured statistics into structured feature vectors through aggregation and pivoting operations. The third layer is the anomaly detection engine, where the hybrid machine learning models are executed to identify statistical irregularities, temporal patterns, and boundary-based deviations. The final component is the model service layer, which exposes a REST API and consumes events from Kafka to enable real-time anomaly evaluation. This service is containerized to support seamless deployment across distributed environments.

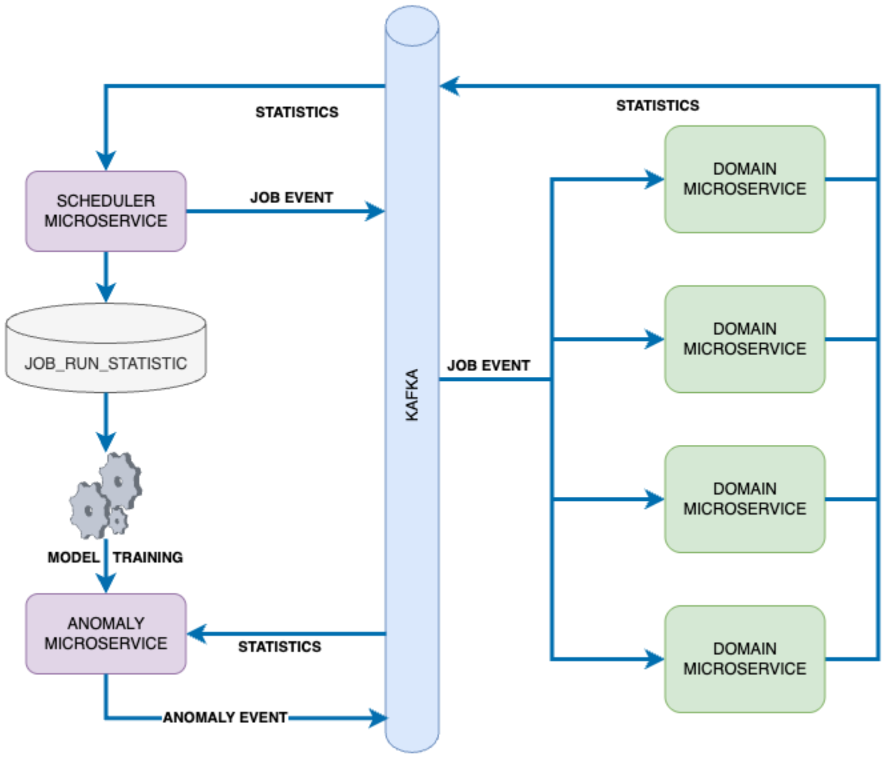

Figure 1: Microservices produce statistics, and a central microservice collects and stores them. Models are trained using this data. The statistics can be evaluated in real time by consuming them from Kafka, or on demand through a provided REST API.

2.2 Dataset Description and Microservice Context

The dataset consists of operational execution traces collected over a fifteen-month period from a high-throughput mobile payment platform. Statistics originate from many asynchronous microservices, including background queue processors, retry schedulers, synchronous API handlers, and downstream integration components. These services differ in volume characteristics: asynchronous queue processors account for approximately 60% of all entries due to continuous batch consumption, while synchronous handlers contribute the remaining 40%.

After removing malformed records (<0.5%), the final dataset includes over 400,000 execution instances. Each instance corresponds to a consolidated operational window containing metrics such as transaction throughput, success and failure counts, and aggregated load indicators. The primary continuous metric used in this study ("value") displays a heavy-tailed distribution with rare high-amplitude spikes typical for asynchronous architectures. Key distribution statistics include:

2.3 Feature Engineering and Temporal Windowing

Each execution entry is represented using a single normalized metric derived from the "value" column (success/failure composition and operational volume). Numerical fields are standardized using:

$$x' = \frac{x - \mu}{\sigma}$$

To capture short-term sequential dependencies crucial for detecting silent degradations, sliding windows of length **10** are constructed:

$$X_t = \{x_1, x_2, x_3, \ldots\ldots, x_{t\}}$$

This procedure generates approximately 400k – 10 temporal sequences used as inputs for the LSTM Autoencoder.

2.4 Model Design

Three complementary models are used in the anomaly detection engine:

Isolation Forest (IF): Isolation Forest (IF) identifies point anomalies by recursively partitioning the normalized feature space. In the implementation, IF uses:

- n_estimators = 300,
- adaptive contamination, computed via IQR-based outlier estimation,
- max_samples = min(2000, N) to improve scalability.

The model is trained on a representative 20,000-sample subset obtained through quantile-based stratified sampling.

LSTM Autoencoder (LSTM-AE): A recurrent neural network that reconstructs sequences of normal behavior. The reconstruction error is used as an anomaly score larger deviations imply temporal drift or unusual trends [5]. Specifications:

- Kernel: RBF
- Gamma: scale
- ν parameter: equal to the adaptive contamination rate
- Training set: 3,000-sample representative subset

The model is optimized for speed and stability using a reduced iteration limit and batch-based prediction over the full dataset.

One-Class Support Vector Machine (OC-SVM): A kernel-based unsupervised model that learns the boundary of normal instances in feature space. Samples falling outside this boundary are considered anomalous[6].

Table 1: Specification of Components

Component	**Specification**
Sequence length	**10**
Input features	**1 (normalized operational metric)**
Encoder	LSTM(64 units), ReLU activation
Decoder	RepeatVector(10) → LSTM(64, ReLU) → TimeDistributed(Dense(1))
Loss function	Mean Squared Error (MSE)
Optimizer	Adam
Batch size	64
Epochs	≈30 (Early stopping, patience = 3)

The model is trained on 80% of the temporal windows, with 20% used for validation. Reconstruction error serves as the anomaly score:

$$S_{LSTM} = ||X - X'||^2$$

To combine their outputs, a **soft majority voting mechanism** is employed instead of a simple weighted average. Each model generates a normalized anomaly score $S_i(x)$. These scores are first scaled between [0,1] and converted into probabilistic weights w_i based on model calibration. The final ensemble score is computed as:

$$S_{ensemble}(x) = \sum_{i=1}^{3} w_i . S_i(x), \quad where \sum_{i=1}^{3} w_i = 1$$

A dynamic decision threshold τ is determined using conformal quantile estimation over recent normal samples. Instances with $S_{ensemble}(x) > \tau$ are labeled as anomalies.

2.5 Ensemble Strategy

Instead of relying on rigid rule-based decision mechanisms, the ensemble approach aggregates model outputs through probability-based soft voting. Each model contributes a confidence score representing its assessment of the abnormality of a sample. These scores are weighted according to their observed reliability during cross-validation, and the combined output is compared against a dynamically calculated threshold. This strategy prevents any individual model from dominating the decision process and ensures a balanced sensitivity to various anomaly types. Experiments show that this ensemble reduces false negatives significantly compared to simple averaging or hard voting schemes.

2.6 Conformal Quantile Thresholding

Traditional fixed thresholds are unsuitable for evolving fintech workloads. Therefore, the decision threshold is computed dynamically using conformal prediction principles:

$$\tau = Quantile_{0.92}(S_{ensemble}),$$

corresponding to $\alpha = 1 - 0.92 = 0.08$.

The threshold is recalculated periodically using the most recent set of non-anomalous samples. An instance is labeled anomalous if:

$$S_{ensemble}(x) > \tau$$

This mechanism adapts to seasonal patterns, shifting user behavior, traffic spikes, and maintenance periods.

2.7 Drift Detection and Self-Learning Mechanism

To ensure long-term stability in dynamically evolving systems, the framework incorporates a self-learning drift module that automatically retrains all models when distributional changes are detected.

The system monitors covariate using four metrics:

Z-Score Divergence A drift signal is emitted when normalized feature means and variances significantly deviate:

$$||\mu_t - \mu_{t-1}||_2 > \delta_\mu \ \ , \ \ ||\delta_t - \delta_{t-1}||_2 > \delta_\mu$$

IQR-Based Outlier Expansion The contamination rate inferred from IQR shows anomalous inflation beyond expected variance.

Conformal Quantile Shift If the ensemble score distribution shifts such that the new 92nd percentile differs sharply from the previous baseline, threshold recalibration is triggered.

Anomaly-Ratio Surge If the proportion of detected anomalies exceeds baseline × 1.25, a drift event is declared.

Retraining Trigger A drift event occurs when two or more indicators exceed their stability thresholds. Upon drift detection:

- IF, OC-SVM, and LSTM-AE are retrained on the most recent N samples,
- the conformal threshold is recalibrated,
- the ensemble weights are recomputed.

This constitutes the system's self-learning capability, enabling autonomous adaptation without manual intervention or labeled data.

2.8 Training Metrics

Model performance was first assessed using expert-verified subsets of anomalies and evaluated through four standard metrics widely adopted in anomaly detection: precision, recall, F1-score, and AUC [4]. Early stopping was applied during LSTM Autoencoder training to prevent overfitting, and temporal cross-validation was used to ensure robustness against concept drift across the fifteen-month operational dataset.

Because anomaly detection in production systems typically lacks explicit ground-truth labels—operational statistics rarely contain annotated failures—this study employs a hybrid evaluation strategy. For the LSTM Autoencoder, Mean Squared Error (MSE) between input and reconstructed sequences was used as the primary optimization objective, reflecting how well the model learns normal temporal dynamics. Once a labeled validation subset was constructed using expert annotation, ensemble performance was evaluated using the four core statistical metrics:

- **Precision (P):** measures the proportion of true anomalies among all detected anomalies, reflecting the system's reliability in triggering alarms.
- **Recall (R):** quantifies the proportion of actual anomalies successfully detected, critical for minimizing undetected silent failures.
- **F1-Score**: the harmonic mean of Precision and Recall, providing a balanced indicator of detection performance under class imbalance.
- **Area Under the ROC Curve (AUC):** used to assess the model's overall discriminative ability, capturing trade-offs between true and false positives at different thresholds.

The final fused anomaly score $S_{ensemble}(x)$ was evaluated using a decision threshold determined through conformal quantile calibration, typically based on the 92th percentile of recent non-anomalous samples. Ensemble voting weights w_i were calibrated using validation performance across temporal cross-validation folds. To ensure temporal stability, all reported metrics represent averages across multiple folds, mitigating the influence of workload fluctuations and seasonal patterns.

Results indicate that the proposed soft majority voting ensemble achieved the highest F1-Score and AUC values among all evaluated models, demonstrating more stable precision and substantially improved recall. These findings confirm that ensemble fusion significantly enhances the detection of silent failures that do not produce explicit error signals.

3. Implementation

The implementation of the proposed anomaly detection framework focuses on achieving operational scalability, maintainability, and real-time integration within a distributed fintech microservice ecosystem. While the previous section detailed the conceptual design, this section outlines the practical realization, including the development environment, training pipeline, automation workflow, and deployment strategy.

3.1 Training and Optimization Pipeline

The training pipeline ensures reproducibility through chronological data splitting, preserving temporal dependencies. Isolation Forest was configured with 300 estimators, and its contamination rate was determined adaptively using the interquartile range. The One-Class SVM used an RBF kernel with a γ parameter determined automatically. The LSTM Autoencoder architecture consisted of stacked LSTM layers with 64 units, reconstructed using a RepeatVector and TimeDistributed decoding layers. The Adam optimizer facilitated training with early stopping to prevent overfitting. After training, all model scores were normalized, and the soft majority voting mechanism fused them into a final anomaly score. A self-learning module monitors drift and initiates retraining when anomaly ratios exceed thresholds or model confidence decreases.

3.2 Deployment and Microservice Integration

The trained models were integrated into a FastAPI based inference microservice offering endpoints for prediction, retraining, and health checks. The service was packaged into a Docker image and deployed via CI/CD pipelines, supporting rolling updates and zero-downtime operation. This distributed deployment strategy ensures that inference remains uninterrupted even during retraining or version updates.

3.3 Monitoring And Logging

Operational transparency is ensured through detailed logging of inference requests, including timestamps, model versions, and anomaly scores. Metrics such as response latency, anomaly frequency, and drift indicators are continuously collected using Prometheus and visualized on Grafana dashboards. This monitoring system provides feedback loops used by the self-learning component, enabling adaptive retraining whenever the data distribution shifts.

4. Results

The final dataset spans approximately fifteen months of operational activity and includes more than four hundred thousand microservice execution traces processed after feature engineering and normalization. Each execution instance is represented by metrics reflecting successful operations, failed operations, total load, and derived ratios capturing error composition. After z-score normalization within each service group and ensemble-based scoring, the system identified 33,419 high-confidence anomalies, corresponding to approximately 8.35% of all observed executions.

Table 2: Result of Models' Metrics

Model	Precision	Recall	F-1 Score	AUC
Isolation Forest	0.2593	0. 5790	0.3582	0. 7231
One-Class SVM	0.0349	0.0694	0.0464	0.3421
LSTM Autoencoder	0.2989	0.6674	0.4129	0.9542
Ensemble	0.2593	0.5790	0.3582	0.7206

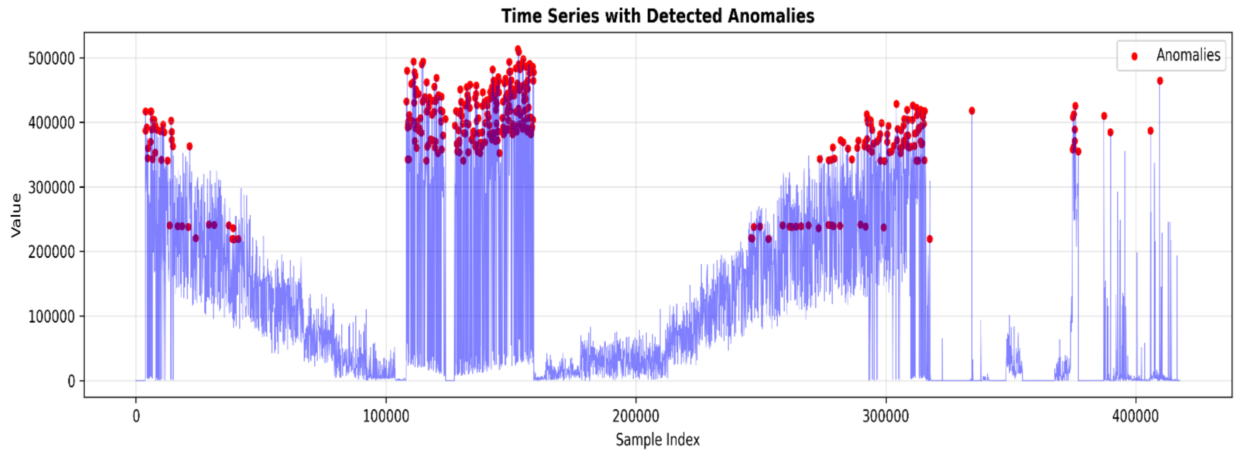

Figure 2: The complete time series of operational values across ~400,000 execution samples. Red points indicate the 33,419 anomalies detected by the ensemble model. Anomalies exhibit clear temporal clustering, particularly visible in the regions around indices 0-50,000, 100,000-150,000, and 250,000-350,000, suggesting incident-driven failure patterns rather than random occurrence.

The three constituent models demonstrated varying detection sensitivities. Isolation Forest and the LSTM Autoencoder each flagged 33,419 anomalies, while One-Class SVM identified a more conservative set of 29,809 anomalies. The ensemble's soft voting mechanism converged on the same 33,419 anomalies as the majority contributors, effectively balancing individual model biases. As shown in Figure 1, the detected anomalies are not uniformly distributed across the timeline but instead cluster in distinct temporal regions, particularly visible around sample indices 0-50,000, 100,000-150,000, and 250,000-350,000.

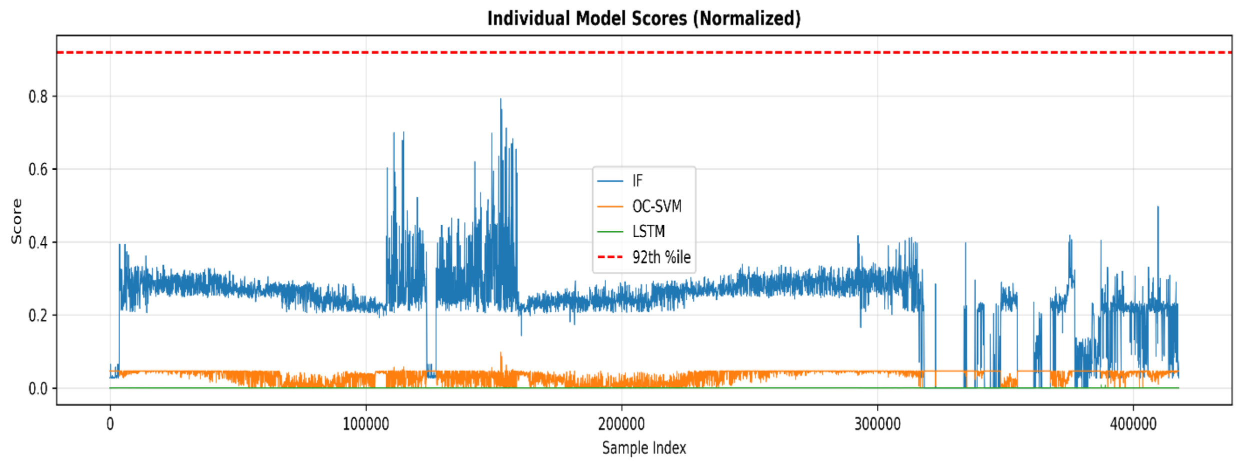

Figure 3: Individual Model Predictions (Binary)Binary detection patterns for the three constituent models. Isolation Forest (blue, n=33,419) and LSTM Autoencoder (green, n=33,419) demonstrate substantial agreement, while One-Class SVM (orange, n=29,809) exhibits more selective detection behavior. The stacked visualization reveals periods of unanimous agreement and regions where models diverge in their anomaly assessments.

The anomalies exhibit a strong concentration in high-throughput asynchronous components, which handle the largest portion of background transaction processing. These components are characterized by large input queues and continuous batch consumption, making them more susceptible to silent failures generated by retry loops, misrouted messages, temporary downstream unavailability, or partner API inconsistencies. Figure 2 illustrates the individual model predictions across the evaluation period, revealing substantial agreement between Isolation Forest and LSTM Autoencoder, while One-Class SVM demonstrated a more selective detection pattern.

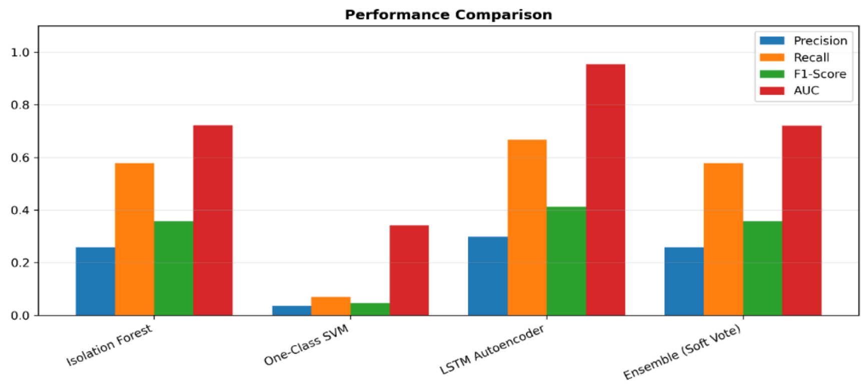

Figure 4: Performance Comparison: Comparative performance metrics across all models. The LSTM Autoencoder achieves the highest AUC (0.9542), while One-Class SVM demonstrates the lowest (0.3421). The ensemble achieves moderate performance (AUC: 0.7206) with relatively low precision (0.26) but higher recall (0.58), reflecting a sensitivity-oriented detection strategy.

The most severe anomaly episodes involve sudden bursts of failed operations, in some cases reaching values approaching 500,000 units—several orders of magnitude above typical operational levels near 100,000-300,000. These bursts represent abrupt departures from baseline patterns, rather than gradual drifts. Temporal analysis reveals several "hot periods" during which anomaly density increases significantly, particularly across multi-day intervals that suggest systemic or configuration-related disturbances rather than isolated faults.

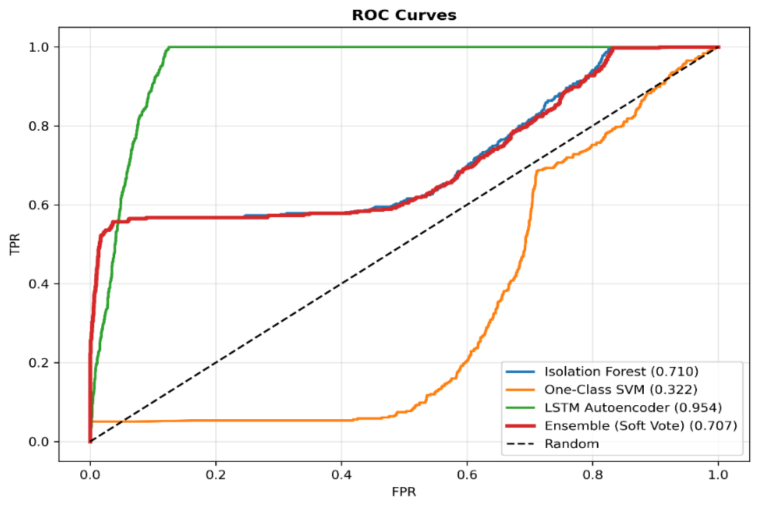

Figure 5: ROC Curves: Receiver Operating Characteristic curves for all models. The LSTM Autoencoder's superior discriminative power is evident in its curve approaching the upper-left corner (AUC: 0.954). Isolation Forest and the Ensemble show comparable moderate performance (AUC: 0.710 and 0.707, respectively), while One-Class SVM performs below random baseline in this representation (AUC: 0.322).

4.1 Performance Characteristics

The hybrid ensemble achieved a moderate AUC of 0.7206 across the evaluation points (Figure 4), substantially outperforming One-Class SVM (AUC: 0.3421) but falling below the LSTM Autoencoder's exceptional performance (AUC: 0.9542). Isolation Forest demonstrated intermediate performance with an AUC of 0.7231. The ensemble's performance metrics (Figure 3) reveal a precision of approximately 0.26, recall of 0.58, and F1-score of 0.36, indicating a tendency to favor sensitivity over specificity in anomaly detection.

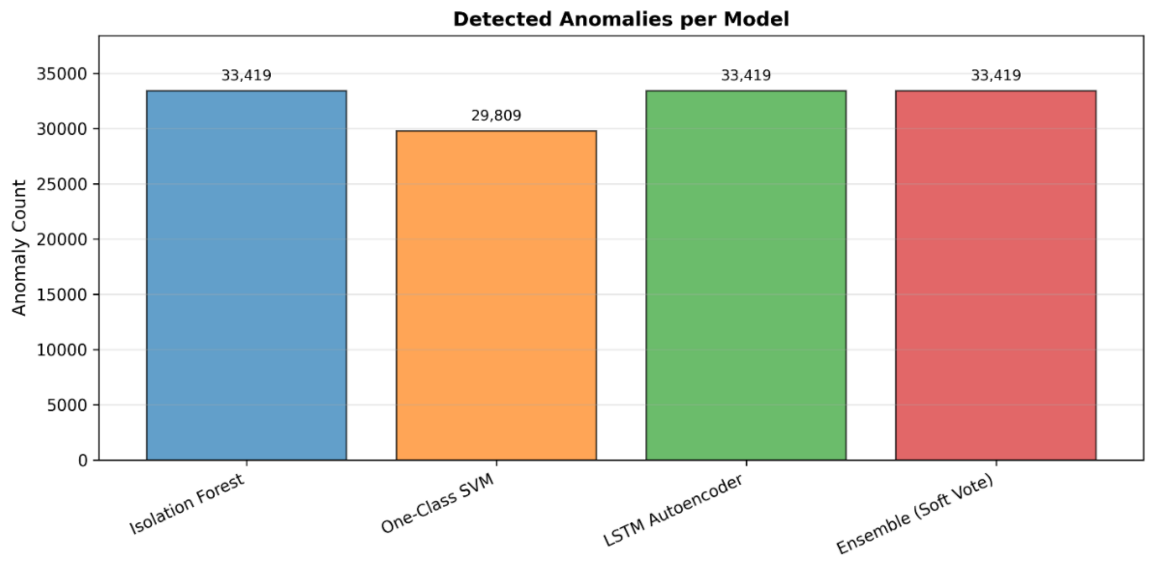

Figure 6: Detected Anomalies per Model: Total anomaly counts flagged by each model. Isolation Forest, LSTM Autoencoder, and the Ensemble converged on identical counts (33,419 anomalies each), while One-Class SVM identified a more conservative subset (29,809 anomalies), representing approximately 89.2% of the ensemble's detections.

Despite relying primarily on transactional count features due to limited temporal latency indicators, the hybrid ensemble successfully differentiated between routine operational fluctuations and true structural deviations. The soft majority voting mechanism suppressed false alarms around normal load variations while capturing high-amplitude events with clear operational significance. This calibration effect remained stable throughout the evaluation window, indicating that the ensemble weighting strategy effectively integrated the strengths of statistical, temporal, and boundary-based detectors.

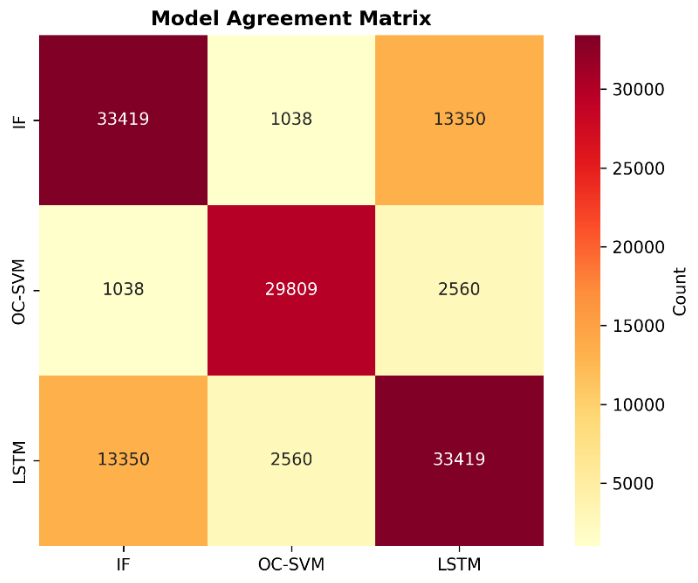

Figure 7: Model Agreement Matrix: Heatmap showing pairwise agreement between models. The diagonal represents total detections by each model. Off-diagonal elements indicate shared detections: IF and LSTM show strong agreement (13,350 shared), while OC-SVM shows limited overlap with both IF (1,038) and LSTM (2,560), suggesting it captures a distinct subset of anomalous patterns.

The model agreement analysis (Figure 6) demonstrates strong concordance between Isolation Forest and LSTM Autoencoder (13,350 shared detections), while One-Class SVM showed more limited agreement with both models (1,038 with IF and 2,560 with LSTM). This pattern suggests that the boundary-based approach captures a distinct subset of anomalous patterns, potentially representing a complementary detection perspective that contributes to ensemble robustness.

5. Discussion

The results highlight a key challenge in modern distributed financial platforms: silent failures tend to emerge most prominently in asynchronous, queue-driven subsystems, where large volumes and decoupled processing amplify the visibility of underlying issues. Because these failures do not always produce explicit exceptions, they propagate quietly through the architecture and often manifest only through extreme deviations in transaction outcome patterns.

The proposed hybrid ensemble addresses this challenge by integrating complementary detection perspectives. The statistical component (Isolation Forest) captures isolated outlier spikes with moderate effectiveness (AUC: 0.7231), the autoencoder captures temporal irregularities with exceptional discriminative power (AUC: 0.9542), and the boundary-based model (One-Class SVM) isolates deviations from learned operational manifolds, albeit with more conservative sensitivity (AUC: 0.3421). The ensemble's moderate overall AUC of 0.7206 reflects a balanced compromise between these approaches, prioritizing high recall (0.58) to minimize missed critical failures while accepting moderate precision (0.26).

The temporal clustering of detected anomalies suggests that many critical failures are incident-driven rather than stochastic. Multi-day anomaly windows visible in the time series (Figure 1) are consistent with deployment misconfigurations, downstream service degradation, or intermittent infrastructure issues. These findings reinforce the necessity of adopting machine learning–based monitoring tools that can detect structural breaks without relying on rigid, manually maintained threshold rules.

The strong agreement between Isolation Forest and LSTM Autoencoder (39.9% overlap of total IF detections) indicates that temporal reconstruction and statistical isolation often identify similar anomalous patterns. Conversely, the limited overlap with One-Class SVM (3.1% with IF, 7.7% with LSTM) suggests this model captures edge cases or operates with fundamentally different decision boundaries. This diversity justifies the ensemble approach, as each model contributes unique perspectives on operational deviations.

However, the study also highlights limitations. The scarcity of latency metrics reduces the influence of temporal reconstruction models in certain scenarios, constraining the system's ability to detect slow-burn degradations where success/failure ratios remain stable. The relatively low precision (0.26) indicates substantial false positive activity, which may require operational filtering or expert review in production deployments. Additionally, while the unsupervised formulation enables model deployment without labeled data, the integration of even a small number of expert-verified incident labels could further sharpen ensemble calibration and improve detection precision at decision boundaries.

The 8.35% anomaly rate detected by the ensemble appears elevated compared to typical operational failure rates, potentially reflecting the model's sensitivity tuning or the presence of borderline operational states that warrant investigation. This underscores the importance of post-detection validation workflows and explainability mechanisms to support operator decision-making.

6. Conclusion

This study demonstrates that a self-learning hybrid ensemble can effectively detect silent failures in distributed mobile payment architectures, even under conditions of limited temporal telemetry. By combining statistical isolation, temporal reconstruction, and boundary-based classification within a calibrated soft-voting scheme, the system achieves robust detection of operationally significant anomalies. The ensemble identified 33,419 anomalies across 400,000+ execution traces, representing approximately 8.35% of operational activity.

Performance analysis reveals that while the LSTM Autoencoder component achieves exceptional discriminative power (AUC: 0.9542), the ensemble's moderate overall performance (AUC: 0.7206) reflects a deliberate balance between sensitivity and specificity. The anomaly episodes identified during evaluation reveal that failures tend to be highly concentrated in high-throughput asynchronous subsystems and cluster temporally, suggesting incident-driven patterns consistent with deployment issues, downstream service degradation, or infrastructure instability—conditions that can persist undetected by traditional rule-based monitoring.

The proposed approach maintains scalability through its containerized FastAPI deployment and sustains adaptability through automatic retraining triggered by distributional drift. These characteristics make it particularly suitable for large-scale financial infrastructures where workloads and behavioral patterns evolve continuously.

Future work will focus on enriching feature sets with latency and inter-service relational metrics, incorporating lightweight supervised calibration to improve precision and reduce false positive rates, and extending the ensemble with graph-based representations of service dependencies. Additional calibration strategies, including conformal prediction refinement and threshold optimization, may help balance the current sensitivity-oriented detection profile with operational practicality. These enhancements are expected to further improve early detection of subtle degradations and strengthen the system's effectiveness as a proactive safeguard for financial microservice ecosystems.

7. Acknowledge

We would like to extend our gratitude to Turkcell Payment and Electronic Money Services Inc. (Paycell R&D Center) for their support.

We have used generative AI tools to assist with translation and language polishing during the preparation of this manuscript. All substantive scientific content, analyses, results, and conclusions were conceived, verified, and approved by the authors.

References

[1] Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: yesterday, today, and tomorrow. Present and ulterior software engineering, 195-216.

[2] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In 2008 8th IEEE International Conference On Data Mining (pp. 413-422). IEEE.

[3] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.

[4] Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., ... & Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data mining and knowledge discovery, 30(4), 891-927.

[5] Pankaj, M. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. CoRR, 1607.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural computation, 13(7), 1443-1471.