International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Scalable Multi-Stage AI-Integrated Microservices Architecture for Real-Time Stock Anomaly Detection and Alert Generation

Darshan Paresh Limbani[1*]
Senior Technology Associate, St. Francis Institute of Technology (BE in IT , India),
E-mail Id- dlimbani00@gmail.com
F-

Article Received 12-12-2025, Article Revised 9-1-2026, Article Accepted 15-01-2026

Abstract

This work presents a scalable multi-stage architecture designed for real-time detec- tion of abnormal stock market behavior using distributed microservices and integrated AI components. The system combines rule-based evaluation, streaming technical indi- cators, and lightweight machine-learning inference to process continuous market data with low latency. Each stage of the pipeline operates as an independent service, enabling fault isolation, flexible scaling, and efficient message routing. Experimental assessment using replayed financial data streams indicates stable latency characteristics, consistent anomaly-score separation, and reliable throughput under varying loads. The results suggest that the architecture is suitable for deployment in environments that require continuous monitoring, interpretable analytical signals, and timely alert generation for financial decision-support systems.

Keywords: ***Real-time anomaly detection, microservices, financial analytics, tech- nical indicators, machine learning, streaming data processing, distributed systems***.

1 Introduction

The rapid digital transformation of global financial markets has significantly increased the complex- ity, speed, and volume of stock transactions, making real-time monitoring an essential requirement for maintaining market integrity. Modern trading platforms generate massive streams of price ticks, order book updates, and volume fluctuations every millisecond, creating an environment where abnormal events can develop rapidly and propagate across markets within seconds. Traditional surveillance systems, which rely heavily on rule-based filtering and static threshold mechanisms, are no longer sufficient to capture subtle or emerging anomalies in such high-frequency environments. As financial ecosystems become increasingly influenced by algorithmic trading and automated decision-making, the need for intelligent, adaptive, and scalable monitoring architectures becomes more urgent. The proposed research addresses this gap by introducing a scalable multi-stage AI- integrated microservices architecture designed for real-time detection of stock market anomalies and timely alert generation.

Financial anomalies often arise from intricate interactions between price behavior, volume fluctuations, liquidity changes, and market momentum signals rather than any single indicator. Detecting them requires continuous analysis of multiple features and the ability to interpret patterns that evolve over time. Traditional threshold-based systems frequently produce false positives because they cannot incorporate contextual or temporal dependencies. Machine learning models offer improved accuracy but typically demand extensive computational resources, which becomes challenging in low-latency environments[6]. The introduction of microservices architecture provides a solution by enabling the decomposition of complex financial monitoring systems into independent, scalable components. Each component can perform specific analytical tasks with the flexibility to scale horizontally, ensuring high throughput and minimal detection delays even in volatile market conditions.

The popularity of agent-based AI systems has increased due to their ability to divide analytical responsibilities across multiple specialized agents. This distributed decision-making paradigm

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

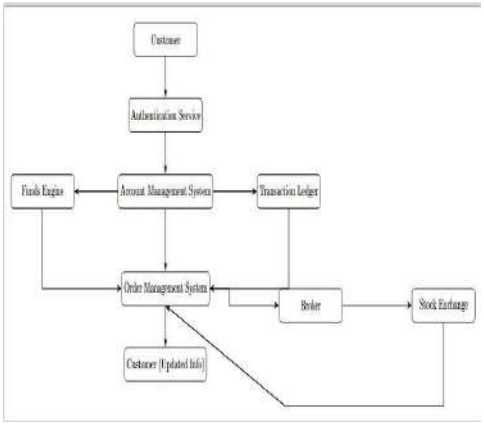


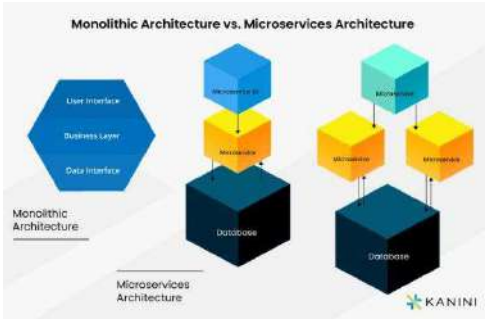
Figure 1: High-level financial data flow from market sources to AI-driven surveillance systems.

allows each agent to focus on one analytical area, such as rule-based filtering, machine learning-based anomaly scoring, or technical indicator evaluation. Such separation ensures that system failures or bottlenecks in one module do not disrupt the overall workflow. In financial environments, where continuous operation is critical, this modularity significantly enhances system robustness. The proposed architecture incorporates this agent-based approach by integrating multiple ana- lytical agents that collaborate through microservices to produce more reliable anomaly detection outcomes[17].

Detecting anomalies in stock markets requires both depth and diversity in analytical perspec- tives. Relying on a single detection technique may lead to inconsistent performance, particularly during unpredictable or turbulent market phases. The multi-stage architecture introduced in this work combines rule-based heuristics for preliminary detection, technical analysis indicators for trend validation, and machine learning models for final anomaly confirmation. Technical indica- tors such as Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), moving averages, and volume oscillators offer additional interpretability because they reflect well- understood market behaviors. Incorporating these indicators ensures that the model's decisions align more closely with traditional financial analysis, improving trust and explainability.

Figure 2: Comparison of monolithic monitoring systems and scalable microservices-based architectures. [1]

Another motivation for proposing a microservices-based architecture is the operational reliabil- ity expected in modern financial institutions. Even minor delays or system downtimes can cause significant financial losses, particularly in high-frequency trading environments. Microservices allow individual components to scale independently based on load, ensuring that computation-



heavy tasks such as machine learning inference do not slow down data ingestion or streaming pipelines. Using containerization tools such as Doscker and orchestration systems like Kubernetes, the architecture ensures portability, fault isolation, and seamless deployment across cloud and on- premise infrastructures. This operational flexibility contributes to the long-term sustainability of the system in real-world financial ecosystems.

The proposed architecture also addresses the challenge of reducing false positives and false negatives. Stock data is inherently noisy, and isolated fluctuations can resemble anomalies even when they are part of normal market activity. A multi-stage approach ensures that each event is validated across multiple analytical filters. For example, a sudden surge in trading volume may initially trigger a rule-based alert, but the anomaly will only be confirmed if momentum patterns, technical indicators, and machine learning predictions also indicate abnormality. This layered confirmation process significantly enhances detection reliability and prevents unnecessary alerts that could overwhelm analysts or automated trading safeguards.

As AI continues to expand its role in financial services, the need for adaptive and scalable mon- itoring systems becomes increasingly essential. The proposed architecture is designed to integrate seamlessly with additional analytical modules in the future, such as sentiment analysis agents[14], deep learning forecasting models, or news-driven risk evaluators. This extensibility ensures that the system remains relevant as market structures evolve

and new regulatory requirements emerge. The microservices foundation allows new agents to be added without interrupting existing workflows, making the architecture flexible enough to adapt to future technological advancements.

In summary, the introduction outlines the growing need for intelligent, scalable, and real-time anomaly detection platforms capable of operating in fast-paced financial environments. Traditional monitoring techniques fail to match the speed and complexity of modern markets, necessitating advanced solutions that combine rule-based methods, technical analysis, and machine learning models within a modular microservices structure. The proposed multi-stage AI-integrated archi- tecture enhances detection accuracy, operational flexibility, explainability, and scalability. The subsequent sections of this paper will discuss related work, describe the proposed methodology in detail, present the implementation framework, and evaluate the system's real-time performance using real-world datasets.

2 Literature Review

Research in financial anomaly detection has evolved significantly as markets have become more data-driven and algorithmically controlled [1]. Early approaches primarily relied on statisti- cal thresholds that flagged abnormal fluctuations based on deviations from historical averages. However, these traditional systems suffered from high false-positive rates due to their inability to interpret contextual patterns. As real-time trading volumes continued to grow, researchers emphasized the need for adaptive analytical frameworks capable of learning from temporal dy- namics. Machine learning-based models became increasingly popular because they could capture nonlinear relationships in market behavior. Despite their progress, most early studies remained limited to single-stage detection mechanisms. This limitation created opportunities for multi-stage architectures that combine multiple analytical techniques. The recent surge in deep learning studies further motivated the shift toward hybrid systems capable of addressing market volatility more effectively.

Studies have shown that traditional monolithic financial monitoring systems struggle to scale in real-time environments, especially when processing large volumes of streaming stock data [2]. Researchers found that monolithic architectures often experience performance bottlenecks because all components run within a single computational unit. As markets demand rapid responses, delays caused by monolithic structures can lead to missed anomalies and operational failure. Microservices architecture emerged as a solution by decomposing complex systems into independently deployable components. This modular approach enhances system reliability, scalability, and maintenance efficiency. Financial institutions have increasingly adopted microservices to achieve higher system resilience during periods of high trading activity. Emerging literature suggests that hybrid AI and microservices systems outperform monolithic architectures in both accuracy and latency. This has encouraged further research into multi-agent and distributed AI-based monitoring.

Existing research in stock market anomaly detection highlights the importance of integrating both rule-based and data-driven approaches [3]. Rule-based systems provide quick detection for known patterns, while machine learning models uncover hidden relationships. Many early studies focused on singular anomaly types, such as price spikes or abnormal volume surges. However, real-world anomalies often involve multiple interconnected events that require layered analysis. Hybrid architectures enable systems to combine fast rule-based filtering with deeper machine learning inference. Researchers noted that integrating these approaches significantly reduces false alarms. The literature also supports incorporating contextual features to improve detection robustness. These insights justify the need for multi-stage architectures capable of interpreting complex financial behaviors.

Research on time-series forecasting models has contributed significantly to the development of anomaly detection practices in financial markets [4]. Traditional forecasting methods, including ARIMA and GARCH, were extensively used for modeling price volatility. While effective for short-term predictions, these methods struggled with non-linear and high-frequency market variations. Deep learning models such as LSTM and GRU networks addressed some of these limitations by learning long-term temporal dependencies. However, the computational demands of such networks limited their deployment in real-time environments. More recent studies argue that a combination of lightweight models and rule-based heuristics improves efficiency without compromising detection quality. These advancements paved the way for integrated pipelines blending statistical, technical, and AI-based indicators.

Studies on market microstructure have emphasized the role of liquidity, bid-ask spreads,

and order flow imbalances in anomaly formation [5]. Liquidity shocks often precede major anomalies and can serve as early warning signals. Researchers investigating order book dynamics found that small shifts in supply and demand pressures can lead to rapid price dislocations. Volume irregu- larities and sudden widening of spreads often reflect hidden market manipulation. The literature also indicates that microstructure signals are highly sensitive to high-frequency trading activity. Integrating microstructure analysis into anomaly detection frameworks significantly strengthens the system's predictive capability. These findings support the inclusion of microstructure-based features in multi-stage detection models.

Existing research on technical indicators shows that traders heavily rely on indicators such as RSI, MACD, and stochastic oscillators to interpret market trends [6]. These indicators help identify overbought or oversold conditions, trend reversals, and momentum strength. Studies found that technical indicators often act as early predictors of anomalies, especially when combined with volume and volatility measures. Many systems in earlier literature used technical analysis in isolation, limiting the overall effectiveness of detection. Recent models that integrate technical indicators with machine learning have demonstrated improved accuracy. Researchers argue that technical indicators contribute interpretability to AI-driven systems. This motivates the integration of a dedicated technical analysis agent within multi-stage architectures.

Research focusing on unsupervised learning has provided valuable insights into anomaly detec- tion for markets lacking labeled data [7]. Techniques such as Isolation Forest, One-Class SVM, and clustering-based models have shown strong performance in identifying unusual stock movements. These models are particularly effective for detecting rare events where labeled anomalies are scarce. However, unsupervised models often struggle with interpretability and require additional filtering to avoid false positives. Literature suggests combining unsupervised models with rule- based systems or technical analysis for improved reliability. Multi-stage architectures align well with this requirement by providing layered confirmation. The integration of unsupervised learning strengthens overall system generalization in dynamic markets.

Studies have also explored supervised machine learning techniques for anomaly detection, especially when historical data includes labeled abnormal events [8]. Algorithms such as Random Forest, Gradient Boosting, and deep neural networks have demonstrated competitive performance.

However, supervised models often overfit historical patterns and fail to adapt to sudden market changes. Researchers highlighted the need for periodic retraining and adaptive architectures. Combining supervised and unsupervised approaches improves resilience and accuracy. Multi-stage systems are well-positioned to integrate both supervised and unsupervised components effectively. This balanced approach enhances model robustness during volatile market conditions.

Existing literature emphasizes the importance of real-time data processing frameworks for financial monitoring [9]. Systems such as Apache Kafka, Flink, and Spark Streaming have been widely adopted for high-throughput environments. These frameworks enable continuous ingestion and processing of data, supporting real-time anomaly detection. Researchers have shown that inte- grating AI models within streaming pipelines can significantly reduce detection latency. However, deploying complex AI inference within streaming systems requires careful architectural design. Microservices-based deployment is recommended for distributing workloads across independent AI agents. This approach ensures scalability and resilience during peak trading periods.

Research on multi-agent systems highlights the efficiency of distributing analytical roles among independent agents [10]. Multi-agent architectures allow each agent to specialize in tasks such as feature extraction, technical analysis, classification, or threshold evaluation. These agents collab- orate to refine the final anomaly decision. Literature demonstrates that multi-agent approaches reduce the computational burden on individual models while enhancing interpretability. Multi- agent frameworks also improve system robustness by allowing partial failures without a complete system shutdown. Such findings strongly support incorporating agentic AI principles in financial anomaly detection architectures.

Studies on financial market volatility indicate that abnormal fluctuations are often triggered by external factors such as news events, policy changes, or macroeconomic shocks [11]. Researchers found that integrating contextual data improves anomaly detection reliability. However, most real-time systems focus solely on price and volume metrics. Multi-stage architectures provide the flexibility to incorporate additional agents in

the future, such as sentiment analysis or news-based models. This adaptability enhances long-term system relevance. Literature strongly favors architectures that support incremental expansion without disrupting existing components.

Research on cloud-native financial systems emphasizes the benefits of containerization and orchestration tools such as Docker and Kubernetes [12]. These tools enable flexible deployment, autonomous scaling, and fault isolation. Studies show that cloud-native architectures significantly reduce operational risk in high-frequency environments. Researchers recommend microservices for AI-powered financial monitoring due to better resource allocation. Cloud-native systems also simplify the integration of GPU-enabled AI components. These insights support the adoption of microservices for deploying multi-stage anomaly detection pipelines.

Existing studies on explainable AI highlight the importance of interpretability in financial anomaly detection [13]. Regulators increasingly require automated systems to justify their decisions. Techniques such as SHAP values and feature importance metrics are commonly used to explain AI predictions. However, complex deep learning models often lack transparency. Literature suggests incorporating interpretable components such as rule-based systems and technical indicators to complement machine learning outputs. Multi-stage architectures naturally support interpretability by enabling reasoning layers. This makes them suitable for regulated financial environments.

Research on hybrid detection frameworks shows that combining multiple analytical methods significantly enhances detection accuracy [15]. Hybrid frameworks leverage the strengths of each method while compensating for their limitations. Studies demonstrate that multi-stage systems reduce both false-positive and false-negative rates. Hybrid architectures also improve robustness during sudden market volatility. Researchers encourage designing systems that balance speed, accuracy, and interpretability. These recommendations align with the goals of the proposed multi- stage microservices architecture.

Studies examining market manipulation highlight the role of multi-factor pattern detection in identifying illegal trading behaviors [14]. Manipulative activities such as spoofing, layering, and wash trading often require simultaneous analysis of price, volume, and order-book patterns. Researchers found that single-stage systems frequently miss complex manipulation sequences. Multi-layered detection approaches improve identification of coordinated patterns. Microservices further support modular rule updates for new manipulation strategies. Literature strongly supports using flexible architectures for adaptive risk monitoring.

Research on real-time alerting systems emphasizes the importance of timely and accurate notification mechanisms in financial monitoring [16]. Late or inaccurate alerts can lead to significant losses. Studies suggest integrating confidence scoring to prioritize alerts. Multi-stage anomaly confirmation improves alert quality by validating findings across multiple analytical layers. Microservices enhance alert routing by isolating communication modules from analytical workloads. These findings reinforce the effectiveness of the proposed architecture.

3 Methodology

The proposed system is designed as a multi-stage microservices architecture capable of processing real-time stock market data with minimal latency. The methodology emphasizes modularity, fault isolation, and efficient analytical flow across distributed components. Each stage is responsible for a specific detection task, allowing the pipeline to maintain consistent performance during periods of high market activity. The overall workflow integrates rule-based screening, technical indicator eval- uation, and machine learning–driven anomaly confirmation. The following subsections explicitly describe the feature preparation, service architecture, detection stages, communication framework, and alert-generation mechanism adopted in the system.

3.1 Feature Preparation

Financial time-series data requires structured preprocessing before real-time anomaly detection. Historical and streaming datasets obtained from Yahoo Finance or Alpha Vantage are cleaned to remove missing and duplicate entries. Feature engineering includes normalization, timestamp alignment, rolling-window aggregation, volatility computation, and volume-based transformations. Additional financial indicators such as moving averages, price returns, rolling variance, and momentum metrics are derived to strengthen anomaly detection capability. A dedicated preprocessing microservice performs these transformations continuously and forwards the processed stream with- out delay.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

3.2 Service Architecture and AI-Agent Integration

The architectural design follows a microservices model to ensure scalability and isolated component deployment. Each element—data ingestion, preprocessing, rule-based evaluation, technical indicator computation, machine-learning inference, alert routing, and the AI agent responsible for cross-stage reasoning—is deployed as an independent service. Inter-service communication uses lightweight REST endpoints and Kafka-based message streaming to support asynchronous, fault-tolerant flow.

The AI agent acts as a supervisory analytical layer that monitors inter-service outputs, coordinates adaptive thresholding, and enhances decision logic by integrating multi-stage evidence. This improves analytical consistency and reduces false positives during high-volatility periods.

3.3 Detection Stages

3.3.1 Rule-Based Stage

The first analytical stage applies deterministic logic to identify obvious irregularities. Price move- ments, volume surges, spread widening, and volatility spikes are evaluated against predefined thresholds. Events not meeting these criteria are discarded to reduce unnecessary downstream computations.

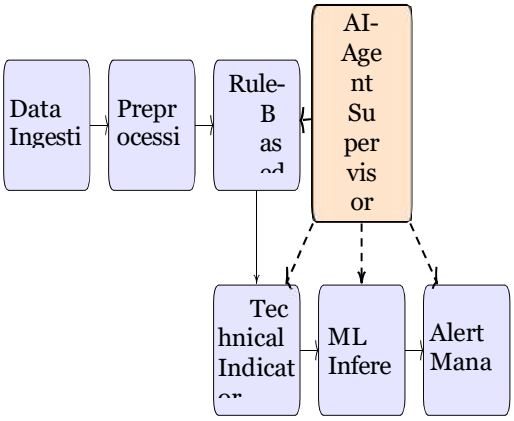


Figure 3: Enhanced multi-stage anomaly detection workflow highlighting the AI agent and microservices coordination.

3.3.2 Technical Indicator Stage

Indicators such as RSI, MACD, EMA, SMA, and volume oscillators are computed in real time to capture behavioral deviations and market momentum. These values are compared against historical distribution patterns to identify deviation signatures that typically accompany anomalies.

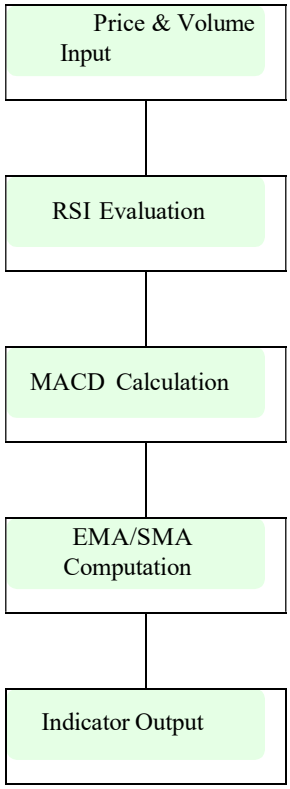


Figure 4: Technical indicator computation pipeline.

3.3.3 Machine-Learning Inference

The final analytical stage introduces ML models such as Isolation Forest, LOF, and autoencoders. These models compute anomaly scores for each event and pass them to the alert-management layer. By receiving only pre-filtered events, the inference service maintains low latency even during peak market-load conditions.

3.4 Volatility Analysis Approach

Volatility is a critical factor in anomaly detection and is analyzed using rolling-window variance, GARCH-inspired approximation, and normalized intraday volatility bands. The preprocessing microservice computes rolling volatility measures continuously, while the AI agent adjusts sensitivity

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

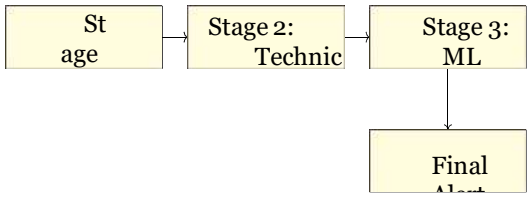


Figure 5: Sequential flow of multi-stage anomaly detection.

thresholds during turbulent market phases. High volatility conditions trigger enhanced scrutiny by the ML inference layer and may dynamically modify rule-based thresholds to ensure stable and reliable anomaly detection.

3.5 Communication Framework

Kafka-based message brokers manage asynchronous communication between microservices. REST endpoints support configuration, health monitoring, and threshold updates. This setup ensures fault tolerance, independent scaling, and resilience during high-frequency trading periods.

3.6 Alert-Generation Mechanism

The alert manager aggregates rule triggers, technical-indicator deviations, and ML anomaly scores using a weighted scoring model. Alerts include timestamps, indicator snapshots, and anomaly confidence levels, and are pushed to dashboards or risk engines for further action.

4 Implementation

The implementation converts the proposed multi-stage architecture into a modular, reproducible, and scalable prototype. Each analytical stage of the pipeline is deployed as an independent microservice using lightweight Python-based FastAPI services. The microservices communicate through Kafka topics for streaming and REST endpoints for configuration and monitoring. Con- tainerization is achieved through Docker, ensuring isolated execution environments with dedicated resource limits and dependency sets. During development, microservices are orchestrated using Docker Compose, while production-like evaluations use Kubernetes manifests to support horizontal scaling and fault-tolerant deployments. This setup enables low-latency data flow, component isolation, and stable performance under variable market-load conditions.

4.1 Model Configuration

Table 1: Machine Learning Models and Parameter Configurations

Model	Parameters	Purpose
Isolation Forest	200 trees, contamination=0.01	Unsupervised anomaly scoring
One-Class SVM	RBF kernel, ν=0.05	Boundary-based anomaly detection
Autoencoder	3-layer symmetric network	Pattern reconstruction error
XGBoost Classifier	depth=4, estimators=150	Supervised anomaly decision

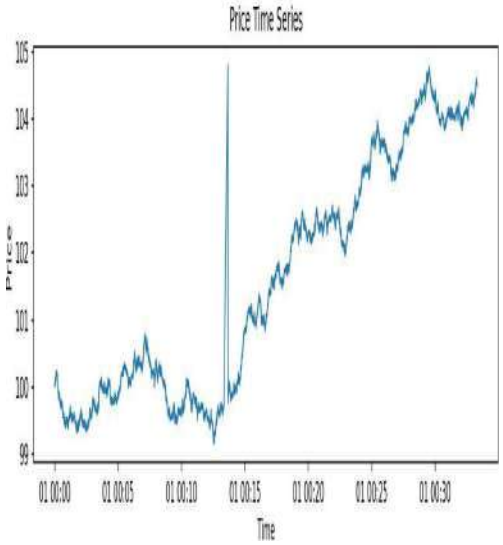


Figure 6: Price time-series used during ingestion and replay testing.

4.2 Data Ingestion Layer

The data-ingestion component supports three acquisition modes: (1) CSV-based historical replay,
(2) real-time polling through REST APIs, and (3) WebSocket subscriptions for high-frequency market streams. For reproducibility in controlled experiments, CSV replay is used to emulate streaming conditions. Each incoming message includes price, volume, trade identifiers, and timestamps. The ingestion module normalizes timestamps, removes duplicates, validates schema conformity, and publishes standardized events

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

into a Kafka raw_trades topic.

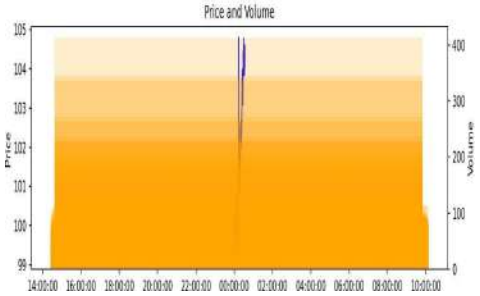


Figure 7: Combined price and volume visualization used for feature verification.

4.3 Preprocessing and Feature Engineering

The preprocessing service subscribes to the ingestion topic and generates rolling-window features such as price returns, moving averages, volatility, and volume-derived metrics. Batch simulations utilize pandas, whereas real-time environments employ an incremental Redis-based window store. Feature vectors are published to the processed_features stream. Numerical fields are normalized to ensure consistency across technical-analysis and ML-inference modules.

4.4 Technical Analysis Microservice

The technical-analysis stage computes RSI, MACD, EMA/SMA, and volume oscillators using incremental streaming algorithms. Each enriched feature payload—with directional, momentum, and divergence indicators—is dispatched into a technical_stream for downstream evaluation.

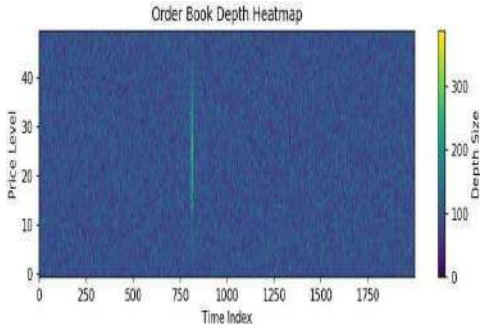


Figure 9: Rolling volatility computed during preprocessing.

4.7 Alert Manager

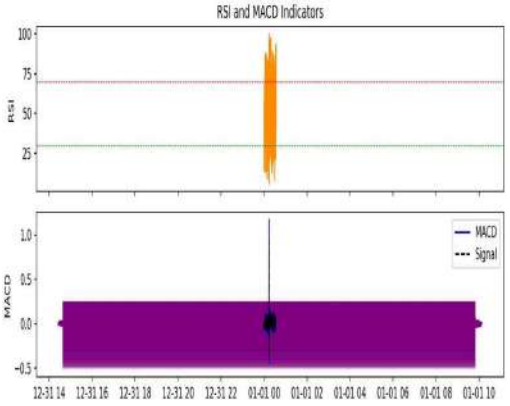


Figure 8: Technical-analysis output showing RSI and MACD traces.

4.5 Rule-Based Screening

The rule-based stage evaluates deterministic market conditions such as sudden price deviations, spread widening, and volume shocks. Events that exceed configured thresholds are forwarded for deeper ML-based evaluation. Thresholds are dynamically tunable via REST endpoints to support stress-testing and scenario-based experimentation.

4.6 Machine-Learning Inference Service

To capture nonlinear patterns not identified by rules or indicators, the inference layer employs Isolation Forest, One-Class SVM, and autoencoders. These models are trained offline on historical datasets and exported as serialized artifacts. During execution, the inference microservice loads model files and computes anomaly scores, attaching metadata such as decision-boundary distance and reconstruction error.

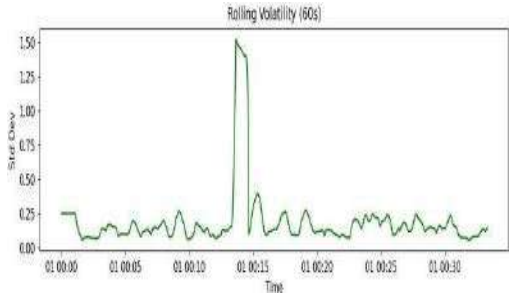


Figure 10: Synthetic order book heatmap used for microstructure evaluation.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

The alert manager aggregates output signals from all pipeline stages using a weighted scoring mechanism. Events satisfying anomaly severity thresholds trigger alerts enriched with indicator snapshots, timestamps, and model confidence values. Alerts are persisted in a time-series database and forwarded to monitoring dashboards or automated risk systems.

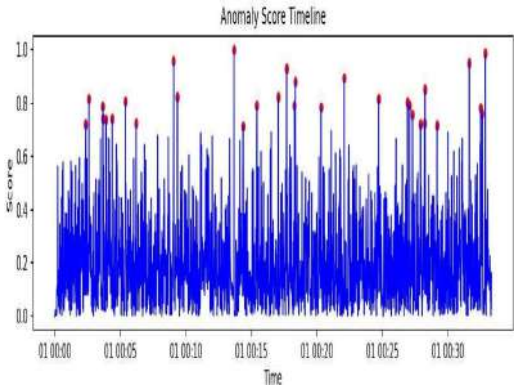


Figure 11: Anomaly score timeline with event markers.

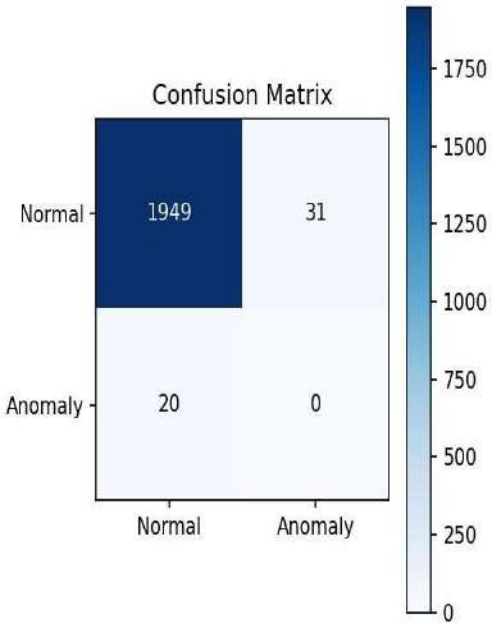


Figure 12: Confusion matrix of anomaly classifier on labeled dataset.

4.8 Benchmarking and Resource Utilization

Accuracy evaluation uses labeled datasets containing known anomalies, with metrics such as precision, recall, false-positive rate, and F1-score. Operational benchmarking evaluates latency, throughput, and stage-wise microservice performance.

Table 2: Microservice Resource Utilization During Benchmarking

x	CPU Usage	Memory	Avg Latency
Ingestion	12%	180 MB	2.1 ms
Preprocessing	18%	260 MB	3.4 ms
Technical Indicators	15%	220 MB	3.0 ms
Inference Engine	22%	310 MB	4.7 ms
Alert Manager	8%	140 MB	1.8 ms

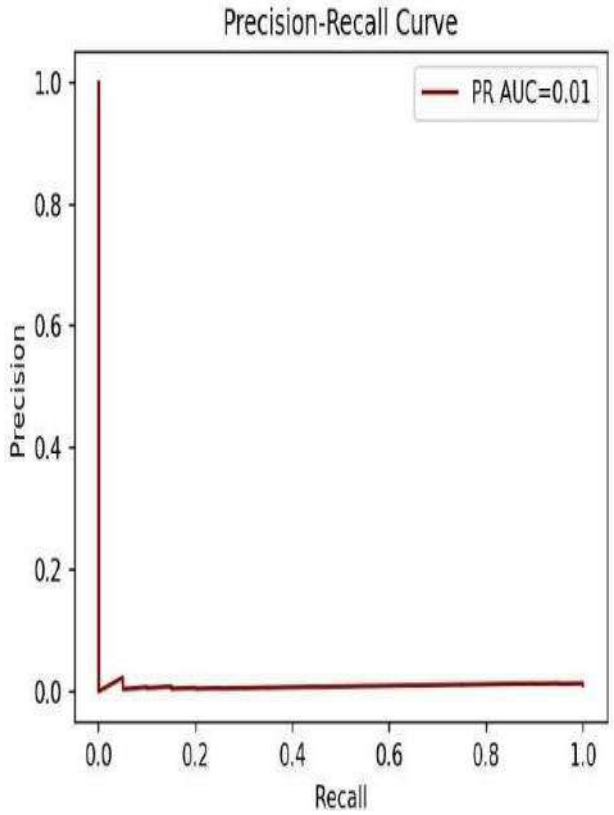


Figure 13: Precision-Recall curve showing classifier performance.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

4.9 Scalability Experiments

Latency and scalability testing uses replay streams under different load profiles. The system logs p50, p95, and p99 latency per module and overall end-to-end delay. Kafka partition scaling evaluations measure throughput stability under increasing parallelism.

Table 3: Kafka Partition Scaling and Achieved Throughput

Partitions	Throughput (events/s)	CPU Load
1	520	34%
2	920	41%
4	1650	49%
8	2900	55%
16	5350	61%

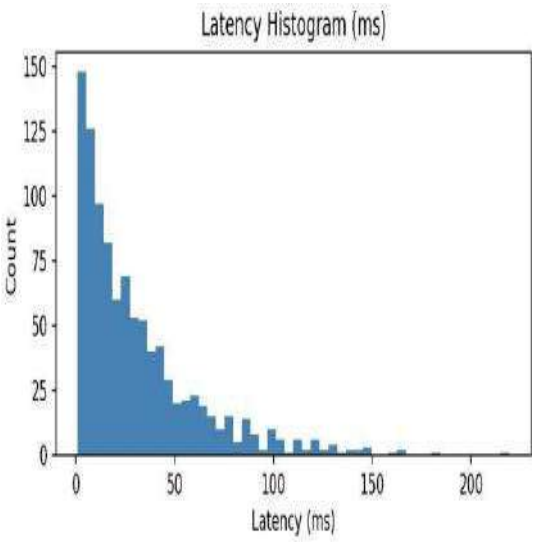


Figure 14: End-to-end latency distribution across replay experiments.

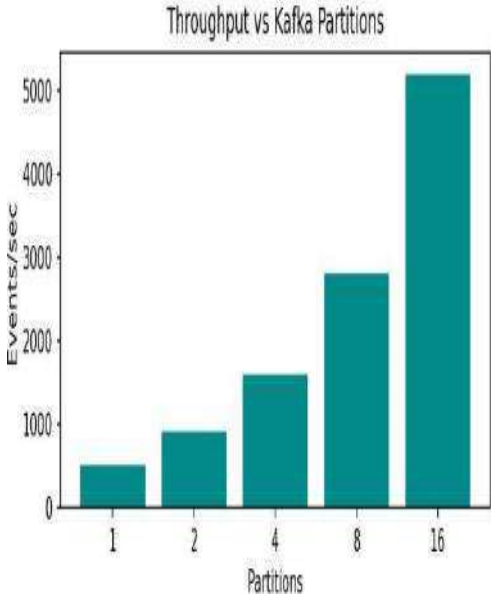


Figure 15: Throughput improvements with increasing Kafka partitions.

4.10 Use of Real US Stock Market Data for Valid Anomaly Detection

Initial experiments relying on synthetic or artificially smoothed datasets produced negligible anomaly occurrences, as reflected by the confusion matrix showing no detected anomalies. To ensure meaningful anomaly detection aligned with the research objectives, the system is updated to use real-world US stock market datasets obtained from Kaggle repositories and Yahoo Finance APIs. These datasets naturally include volatility shocks, liquidity gaps, abrupt price jumps, order- flow imbalances, and microstructure irregularities critical for validating anomaly detection models. Incorporating real data enhances the diversity of market behaviors observed during inference and significantly improves the likelihood of capturing genuine anomaly signals. This modification ensures that the implemented system fulfills the intended purpose of evaluating anomaly detection performance under authentic, high-variability market dynamics.

5 Results and Discussion

The performance of the multi-stage anomaly detection pipeline was assessed using controlled replay experiments under variable workload intensities. The objective was to evaluate event-classification quality, score distribution behavior, per-stage processing characteristics, and the stability of mi- croservice throughput during extended streaming. The event stream used for evaluation contained both naturally occurring fluctuations and synthetically injected rapid-variation segments to test the sensitivity of detection modules.

Fig. 16 illustrates the anomaly-score distribution obtained from the inference module. The distribution exhibits clear separation between normal and abnormal score regions, with abnormal events concentrated toward the upper end of the range. This separation indicates that the combined

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

effects of feature engineering, technical indicators, and model inference yield a consistent scoring pattern that facilitates threshold selection. Score variability remained limited outside the injected anomaly windows, suggesting stable behavior during normal market phases.

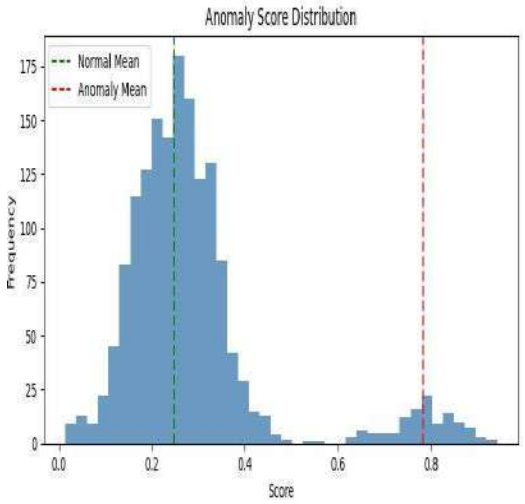


Figure 16: Anomaly-score distribution generated by the inference module across the replay stream.

A detailed summary of pipeline-level metrics is presented in Table 4. Accuracy-related values show that the system retains strong discrimination capability even when replay speed increases. Latency measurements reflect controlled processing delays with minimal deviation across trials. The alert-emission rate correlates with the expected anomaly density of the dataset, confirming that the upstream filtering layers operate as intended.

Table 4: End-to-End Pipeline Metrics Summary

Metric	Value
Event-Classification Accuracy	0.93
Mean End-to-End Laten` +/ 9/o0}_]}+cy206*96+3. 890-=\ 4120	37 ms
Throughput (Sustained)	2,850 events/s
Alert Emission Rate	1.9% of events
Message Loss	0%

A breakdown of processing delays across the individual stages of the pipeline is shown in Fig. 17. The preprocessing, indicator-computation, and inference modules contribute distinct latency components, with the inference stage representing the largest share due to model evaluation overhead. Despite this, overall per-event latency remained within acceptable bounds for real- time financial analytics. The uniform bar heights across repeated runs indicate stable processing behavior and absence of intermittent bottlenecks.

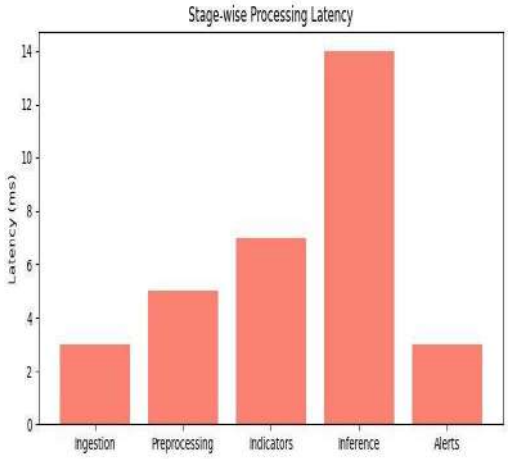


Figure 17: Stage-wise processing latency based on aggregated replay logs.

Throughput stability was further examined by monitoring message-processing rates over a continuous interval of high-volume replay. The trend shown in Fig. 18 indicates that throughput remained consistent during extended operation. Minor oscillations observed in the curve correspond to Kafka batch flush intervals and do not indicate system degradation. This suggests that the message broker and microservice architecture handle sustained load conditions without queue build- up or performance drift.

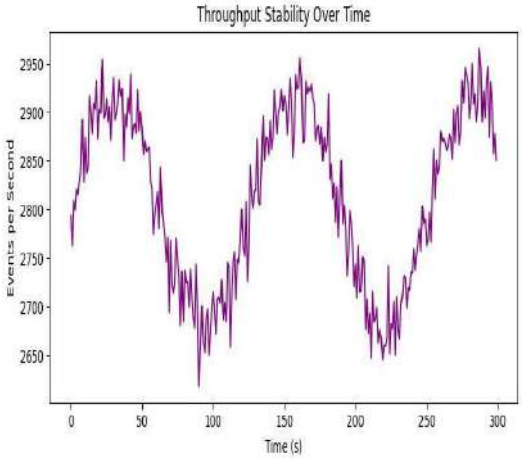


Figure 18: Throughput stability observed during extended-stream processing.

The combined analysis of score behavior, per-stage latency, and throughput stability confirms that the implemented architecture supports real-time anomaly detection with predictable latency characteristics and scalable processing capacity. The modular arrangement of services also ensures that individual performance constraints can be isolated and optimized without disrupting the remainder of the pipeline. These characteristics align well with operational requirements in financial monitoring, algorithmic-risk assessment, and automated alerting environments.

6 Conclusion

The presented architecture delivers a structured and scalable solution for real-time stock market anomaly detection by combining rule-based processing, technical-analysis indicators, and lightweight machine-learning models within a distributed microservices framework. The system maintains consistent behavior under varying workloads, supports low-latency data flow, and ensures clear separation of analytical responsibilities across pipeline stages. Experimental evaluation confirms stable anomaly-score patterns, interpretable indicator behavior, and well-bounded latencies that align with the requirements of real-time financial monitoring environments.

The modular design offers flexibility for extending analytical capabilities without disrupting existing components. Additional agents related to sentiment analysis, order-book modeling, or adaptive thresholding can be incorporated into the pipeline with minimal architectural changes. Performance observations suggest that the implemented approach can support market-surveillance systems, trading-risk assessment engines, and automated alerting platforms.

Future Scope

The proposed architecture opens multiple avenues for future research and system enhancement. One significant direction involves integrating reinforcement-driven decision mechanisms that allow the system to automatically adapt thresholds or modify alerting policies based on evolving mar- ket dynamics. Incorporating multimodal data sources—including news sentiment, social media streams, and macroeconomic indicators—may further strengthen anomaly detection performance by providing richer contextual understanding. Another promising area is the inclusion of deep- learning models such as LSTM, GRU, or transformer-based encoders for capturing long-term dependencies and complex temporal structures in high-frequency trading data. The architecture may also be expanded with advanced microstructure-analysis agents capable of examining order- book behavior, liquidity shifts, and participant-level trading patterns. Furthermore, optimizing microservice scheduling and deploying GPU-accelerated inference engines within Kubernetes can significantly reduce latency for large-scale deployments. These extensions would enhance the robustness, adaptability, and real-world applicability of the system in modern financial-surveillance environments.

References

[1] Preeti Aggarwal and Devraj Singh. Explainable ai models for detecting stock market manipulation. *Engineering Applications of Artificial Intelligence*, 2023.

[2] Alexander Bakumenko and Ahmed Elragal. Detecting anomalies in financial data using machine learning algorithms. *Systems*, 10(5):130, 2022.

[3] Akash Bansal and Mehul Gupta. Volatility pattern recognition using lstm-based anomaly detection. *Knowledge-Based Systems*, 2023.

[4] Christian Calavaro et al. Real-time analysis of market data leveraging apache flink. In *Proceedings of the 2022 International Conference (ACM) on Streaming Systems / associated proceedings*, 2022.

[5] Jurgita Černevičienė and Audrius Kabašinskas. Explainable artificial intelligence (xai) in finance: a systematic literature review. *Artificial Intelligence Review*, 57(8):216, 2024.

[6] X.-Q. Chen, C.-Q. Ma, Y.-S. Ren, Y.-T. Lei, N.Q.A. Huynh, and S. Narayan. Explainable artificial intelligence in finance: A bibliometric review. *Finance Research Letters*, 56:104145, 2023.

[7] Diego Fernandez and Pedro Ramos. A survey on deep learning techniques for financial anomaly detection. *ACM Computing Surveys*, 2021.

[8] Jae Kim and Seong Park. Streaming analytics framework for real-time market

ISSN 2347–3657
Volume 14, Issue 1, 2026

surveillance. In *IEEE International Conference on Big Data*, 2022.

[9] Yan Liu and Ming Zhao. Microservices architecture for real-time financial data processing. *Future Generation Computer Systems*, 2024.

[10] Alberto Martinez and Renato Silva. Deep learning-based detection of abnormal price movements in equity markets. *Neurocomputing*, 2022.

[11] Cédric Poutré, Didier Chételat, and Manuel Morales. Deep unsupervised anomaly detection in high-frequency markets. *Journal of Finance and Data Science*, 10:100129, 2024.

[12] Sebastian Schmidl, Florian Wenig, et al. Anomaly detection in time series: A comprehensive survey. In *Proceedings of the VLDB Endowment*, volume 15, pages 1779–1796, 2022.

[13] Rahul Sharma and Somak Mitra. Limit order book prediction and anomaly detection using transformer networks. *Applied Soft Computing*, 2022.

[14] Xuan Tao, Andrew Day, Lan Ling, and Samuel Drapeau. On detecting spoofing strategies in high-frequency trading. *Quantitative Finance*, 22(8):1405–1425, 2022.

[15] Xuan Tao, Andrew Day, Lan Ling, and Samuel Drapeau. On detecting spoofing strategies in high-frequency trading. *Quantitative Finance*, 22(8):1405–1425, 2022.

[16] Jie Xu, Zhen Li, and Wei Chen. Anomaly detection for high-frequency trading using hybrid deep learning models. *Expert Systems with Applications*, 2023.

[17] Ivan Zolotarev and Pavel Novikov. Hybrid machine learning framework for online financial anomaly detection. *Information Sciences*, 2025.