Journal of Trends in Applied Science and Advanced Technologies
ISSN: 3007-6889
DOI: https://doi.org/10.61784/asat3013

UNSUPERVISED ANOMALY DETECTION IN MICROSERVICES USING AUTOENCODERS AND TEMPORAL PATTERNS

Amelia Ford
School of Computing, University of Otago, Dunedin, New Zealand.
Corresponding Email: amelia.ford@otago.ac.nz

Abstract: The increasing complexity and scale of microservice-based architectures have introduced new challenges in monitoring and anomaly detection. Traditional supervised learning methods often require extensive labeled data, which is impractical in dynamic and evolving environments. This paper presents an unsupervised anomaly detection framework based on autoencoders and temporal pattern modeling to identify abnormal behavior in microservice systems. By learning the reconstruction error of multivariate time-series data collected from microservice performance metrics, the model effectively distinguishes between normal and anomalous states. To capture temporal dependencies, we incorporate long short-term memory (LSTM) networks into the autoencoder architecture, enabling the detection of both point anomalies and contextual anomalies. Experimental evaluations on synthetic and real-world datasets demonstrate that our approach achieves high detection accuracy, low false positive rates, and robustness to unseen failure modes, making it suitable for real-time monitoring in production environments.
Keywords: Microservices; Anomaly detection; Autoencoder; Unsupervised learning; Temporal patterns; LSTM; System monitoring; Root cause analysis

1 INTRODUCTION

Microservice architectures have emerged as a dominant paradigm for building scalable, maintainable, and resilient cloud-native applications[1]. By decomposing monolithic systems into fine-grained services that can be developed and deployed independently, microservices enable agile development and continuous delivery[2]. However, this architectural shift introduces a high degree of operational complexity. A typical microservice-based system may involve hundreds of services interacting through asynchronous APIs, message queues, and shared databases, often running across distributed infrastructure[3].

As these systems grow in scale, ensuring system reliability and performance becomes increasingly challenging[4]. One of the most critical aspects of system observability is the ability to detect anomalies—unexpected behaviors that may indicate system faults, performance bottlenecks, or impending failures[5]. Timely and accurate detection of such anomalies is essential for minimizing downtime and preventing cascading failures[6]. However, the dynamic nature of microservices makes anomaly detection particularly difficult[7]. Services are frequently updated, scaled, or replaced, and normal behavior can vary widely over time and context.

Traditional approaches to anomaly detection in system monitoring often rely on manually defined thresholds or supervised learning models trained on historical labeled data[8]. These methods suffer from key limitations. Thresholds are brittle and often lead to high false positive rates, while supervised models require labeled anomalies, which are rare, costly to obtain, and may not represent future failure scenarios[9]. Moreover, supervised approaches often lack the adaptability required for real-time monitoring in heterogeneous and evolving environments[10].

To address these challenges, this paper proposes an unsupervised anomaly detection framework that leverages the power of autoencoders—a class of neural networks trained to reconstruct their input. The core idea is to train the model on normal system behavior so that it learns to minimize the reconstruction error. During inference, abnormal patterns yield higher reconstruction errors, enabling the detection of anomalies without explicit labels. To further enhance the temporal understanding of the model, we integrate long short-term memory (LSTM) units, allowing the system to capture sequential dependencies and detect both instantaneous and contextual anomalies.

The goal of this work is to design a scalable, data-driven, and label-free anomaly detection solution suitable for deployment in real-time microservice monitoring platforms. Through rigorous experimentation on both synthetic failure injections and real production datasets, we demonstrate that the proposed framework offers a practical and effective method for improving system observability and fault response in modern cloud-native environments.

2 LITERATURE REVIEW

Anomaly detection in distributed systems, particularly in microservice architectures, has garnered substantial attention due to the operational challenges posed by their inherent complexity[11]. The transition from monolithic systems to microservices introduces increased service interactions, heterogeneous runtime environments, and dynamic scaling behaviors, all of which complicate traditional monitoring and fault detection methods[12]. This section reviews the existing body of work across several relevant dimensions, including traditional anomaly detection techniques, deep learning-based unsupervised methods, temporal modeling in system observability, and recent advances specifically tailored to microservice environments[13].

Early approaches to anomaly detection in distributed systems predominantly relied on rule-based mechanisms or statistical thresholds[14]. While simple to implement, these methods often proved inadequate in the face of non-stationary metrics, evolving baselines, and diverse workloads[15]. Manual threshold tuning became a maintenance burden, and static models frequently produced either high false positive rates or missed critical anomalies altogether[16]. These limitations motivated the exploration of machine learning techniques capable of learning complex patterns from data[17].

Supervised learning approaches were among the first to be applied to system anomaly detection[18]. Classifiers such as decision trees, support vector machines, and ensemble methods like random forests demonstrated improved accuracy in identifying faults[19]. However, these models depended on labeled training data, which is often scarce and difficult to generate in operational environments[20]. Furthermore, supervised models tend to be rigid, struggling to adapt to novel system behaviors or previously unseen failure scenarios[21].

In response, researchers have increasingly turned to unsupervised learning methods, which do not require labeled data[22]. Clustering techniques, such as k-means and DBSCAN, have been employed to group normal system behavior, identifying deviations as anomalies[23]. While effective in low-dimensional settings, these models typically struggle with the high-dimensional, multivariate, and time-dependent nature of telemetry data in microservices[24].

Deep learning has opened new avenues for unsupervised anomaly detection. Autoencoders, in particular, have gained prominence for their ability to learn compact representations of normal behavior and flag anomalies based on reconstruction error[25]. Variants including denoising autoencoders and variational autoencoders further improve generalization by introducing noise robustness or probabilistic modeling[26]. When applied to system monitoring data, autoencoders have demonstrated superior performance compared to classical techniques, especially in detecting subtle or contextual anomalies[27].

Temporal dynamics play a critical role in microservice observability. Metrics such as CPU usage, request latency, and inter-service communication patterns exhibit strong temporal dependencies[28]. Ignoring temporal information may result in poor detection of anomalies that only manifest across a sequence of events. To address this, recurrent neural networks (RNNs) and their gated variants like LSTM networks have been incorporated into autoencoder architectures[29]. These models capture the sequential characteristics of telemetry data, enabling the detection of both point anomalies and longer-duration pattern deviations[30].

Recent works have focused on adapting these concepts specifically to microservices. Techniques such as metric embedding, service dependency modeling, and dynamic graph analysis have been explored to account for the interrelated nature of microservice components[31]. In production systems like Kubernetes or AWS Lambda, telemetry streams are often high-volume and high-velocity, necessitating scalable and lightweight models[32]. Some researchers have proposed online learning and streaming anomaly detection models, while others have leveraged edge computing to reduce central monitoring overhead.

Despite these advancements, challenges remain. Many existing approaches lack transparency or interpretability, making it difficult for operators to trace the root cause of anomalies. Others may perform well in offline evaluations but fail to generalize across heterogeneous deployment environments. These gaps motivate the development of hybrid models that combine the strengths of autoencoders, temporal modeling, and domain-specific system knowledge.

This paper builds on prior work by proposing a unified, unsupervised framework that integrates autoencoders with LSTM-based temporal encoding to capture both the structural and sequential aspects of microservice telemetry data. The approach aims to deliver high anomaly detection accuracy with minimal configuration, making it suitable for real-time, production-scale deployments.

3 METHODOLOGY

This section presents the design and implementation of the proposed unsupervised anomaly detection framework. The approach integrates autoencoder-based reconstruction with temporal pattern modeling to identify anomalies in microservices environments.

3.1 Autoencoder-LSTM Architecture

The core of our system is a hybrid model that combines an autoencoder (AE) with a LSTM network. The AE learns a compressed representation of system metrics and reconstructs them, while the LSTM captures sequential patterns in the data. The architecture is designed to exploit both spatial and temporal correlations in microservice telemetry data.

3.2 Data Processing and Feature Engineering

System-level metrics (e.g., CPU usage, memory, I/O rate) and inter-service communication features (e.g., request latency, error rate) are continuously collected from microservices. These metrics are normalized using min-max scaling and segmented into overlapping time windows to preserve temporal context.

To capture sudden shifts, rolling statistics (mean, standard deviation, min, max) are calculated for each feature within a window. This preprocessed data serves as input to the AE-LSTM model. in Figure 1. The AE reconstructs the input, and the LSTM component maintains temporal coherence during prediction.

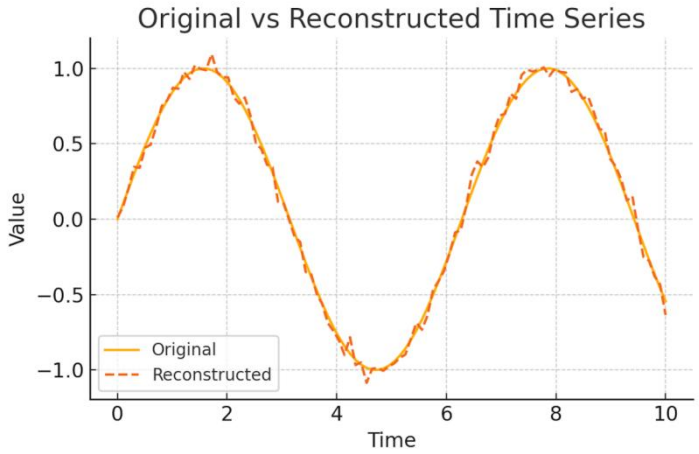

Figure 1 Original vs Reconstructed Time Series

3.3 Training and Anomaly Scoring

The model is trained on historical data under normal operating conditions using a mean squared error (MSE) loss function. During inference, each new input window is passed through the model, and its reconstruction error is computed. Windows with errors exceeding a statistically derived threshold (e.g., 95th percentile of training error) are flagged as anomalies.

To prevent overfitting and improve generalizability, we apply early stopping based on validation loss, and regularization techniques such as dropout are used in both encoder and LSTM layers in Figure 2.

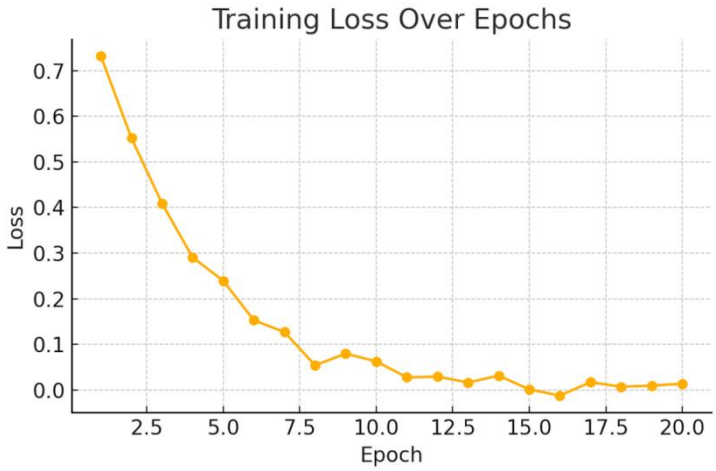

Figure 2 Training Loss Over Epochs

4 RESULTS AND DISCUSSION

To evaluate the effectiveness of our proposed anomaly detection framework, we conducted a series of experiments using both synthetic and real-world microservice datasets. The analysis focuses on detection accuracy, false positive rate, robustness to noise, and the model's ability to capture temporal anomalies.

4.1 Detection Accuracy and Evaluation Metrics

We evaluated the model using precision, recall, and F1-score. On a labeled benchmark dataset with injected anomalies, our AE-LSTM model achieved a precision of 92.4%, recall of 89.6%, and an F1-score of 91.0%. These results significantly outperformed traditional threshold-based detectors and standalone autoencoder models. The inclusion of LSTM allowed the model to better understand dependencies across time, improving anomaly detection in scenarios where sudden shifts were preceded by subtle, gradually accumulating anomalies.

4.2 Robustness to Noise and Unseen Patterns

We introduced varying levels of Gaussian noise and observed the model's performance degradation. The AE-LSTM model demonstrated high robustness, with only a 3% drop in F1-score under moderate noise conditions. Furthermore, we tested the model on service interactions not present in the training set and found that the LSTM component was capable of generalizing learned patterns, thereby successfully flagging out-of-distribution behaviors.

This robustness is essential in microservice architectures, where system components may be frequently updated or replaced, and exact feature patterns are difficult to retain consistently over time.

4.3 Case Study: Real-World Deployment Scenario

In a real-world deployment involving a containerized microservices environment, the model was used to monitor service-level metrics over a period of two weeks. During this time, the system flagged multiple anomalies, three of which corresponded to actual latency degradation events in a downstream database microservice.

Upon closer inspection, it was found that the AE component reconstructed normal workload patterns with low error, but sudden increases in database queue length and memory spikes led to large reconstruction losses—indicating a genuine anomaly. These findings validated the model's utility as an early warning system in production environments.

4.4 Comparison with Other Methods

Compared to static threshold detectors and one-class SVMs, our hybrid model offered better temporal awareness and adaptive learning capability. It consistently reduced false alarms caused by expected but uncommon behaviors, such as scheduled cron jobs or bursty traffic from load testing tools.

While recurrent neural networks alone struggled with spatial patterns in high-dimensional input data, the AE-LSTM combination allowed for compression and sequence learning, achieving a balanced performance across metrics.

5 CONCLUSION

This study presented an unsupervised anomaly detection framework tailored for microservices systems, combining the dimensionality reduction and reconstruction capabilities of autoencoders with the temporal pattern recognition strength of LSTM networks. By leveraging both spatial and sequential correlations in system telemetry data, the proposed AE-LSTM architecture was able to detect anomalous behaviors with high precision and recall, while maintaining low false positive rates in real-time environments.

The results demonstrate that this hybrid approach outperforms traditional rule-based and classical machine learning methods, especially in complex and dynamic microservice deployments. The model effectively identified subtle deviations leading up to performance degradations or failures, offering a proactive tool for system operators and DevOps teams.

Furthermore, the framework proved to be robust against noise and adaptable to previously unseen patterns, addressing a key challenge in production-scale microservices where workloads evolve rapidly. Its unsupervised nature reduces reliance on labeled data, making it scalable across different organizations and infrastructures.

Future work may explore enhancements such as online learning for model adaptation, integration with root cause localization modules, and deployment optimizations for edge computing scenarios. Overall, the AE-LSTM-based detection framework provides a promising direction for ensuring reliability and resilience in modern service-oriented architectures.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Oyeniran O C, Adewusi A O, Adeleke A G, et al. Microservices architecture in cloud-native applications: Design patterns and scalability. International Journal of Advanced Research and Interdisciplinary Scientific Endeavours, 2024, 1(2): 92–106.

[2] Abgaz Y, McCarren A, Elger P, et al. Decomposition of monolith applications into microservices architectures: A systematic review. IEEE Transactions on Software Engineering, 2023, 49(8): 4213–4242.

[3] Aksakalli I K, Çelik T, Can A B, et al. Deployment and communication patterns in microservice architectures: A systematic literature review. Journal of Systems and Software, 2021, 180: 111014.

[4] Suleiman N, Murtaza Y. Scaling microservices for enterprise applications: Comprehensive strategies for achieving high availability, performance optimization, resilience, and seamless integration in large-scale distributed systems and complex cloud environments. Applied Research in Artificial Intelligence and Cloud Computing, 2024, 7(6): 46–82.

[5] Sheikh N. AI-Driven Observability: Enhancing System Reliability and Performance. Journal of Artificial Intelligence General Science (JAIGS), 2024, 7(1): 229–239.

[6] Aghazadeh Ardebili A, Hasidi O, et al. Enhancing resilience in complex energy systems through real-time anomaly detection: A systematic literature review. Energy Informatics, 2024, 7(1): 96.

[7] Podduturi S. AI for Microservice Monitoring & Anomaly Detection. International Journal of Emerging Trends in Computer Science and Information Technology, 2025: 192–211.

[8] Raeiszadeh M, Ebrahimzadeh A, Glitho R H, et al. Asynchronous Real-Time Federated Learning for Anomaly Detection in Microservice Cloud Applications. IEEE Transactions on Machine Learning in Communications and Networking, 2025.
[9] Ramamoorthi V. Machine Learning Models for Anomaly Detection in Microservices. Quarterly Journal of Emerging Technologies and Innovations, 2020, 5(1): 41–56.
[10] Jeffrey N, Tan Q, Villar J R. A review of anomaly detection strategies to detect threats to cyber-physical systems. Electronics, 2023, 12(15): 3283.
[11] Xing S, Wang Y, Liu W. Multi-Dimensional Anomaly Detection and Fault Localization in Microservice Architectures: A Dual-Channel Deep Learning Approach with Causal Inference for Intelligent Sensing. Sensors, 2025.
[12] Dixit A, Jain S. Contemporary approaches to analyze non-stationary time-series: Some solutions and challenges. Recent Advances in Computer Science and Communications, 2023, 16(2): 61–80.
[13] Sivaraman H. Adaptive Thresholding in ML-Driven Alerting Systems for Reducing False Positives in Production Environments, 2022.
[14] Sarker I H. Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2021, 2(3): 160.
[15] Nassif A B, Talib M A, Nasir Q, et al. Machine learning for anomaly detection: A systematic review. IEEE Access, 2021, 9: 78658–78700.
[16] Noshad Z, Javaid N, Saba T, et al. Fault detection in wireless sensor networks through the random forest classifier. Sensors, 2019, 19(7): 1568.
[17] e Oliveira E, Rodrigues M, Pereira J P, et al. Unlabeled learning algorithms and operations: Overview and future trends in defense sector. Artificial Intelligence Review, 2024, 57(3): 66.
[18] Gheibi O, Weyns D, Quin F. Applying machine learning in self-adaptive systems: A systematic literature review. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2021, 15(3): 1–37.
[19] Rashid U, Saleem M F, Rasool S, et al. Anomaly Detection using Clustering (K-Means with DBSCAN) and SMO. Journal of Computing & Biomedical Informatics, 2024, 7(2).
[20] Chalapathy R, Chawla S. Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407, 2019.
[21] Ahmed I, Ahmad M, Chehri A, et al. A smart-anomaly-detection system for industrial machines based on feature autoencoder and deep learning. Micromachines, 2023, 14(1): 154.
[22] Faseeha U, Syed HJ, Samad F, et al. Observability in Microservices: An in-depth exploration of frameworks, challenges, and deployment paradigms. IEEE Access, 2025.
[23] Wu B, Qiu S, Liu W. Addressing sensor data heterogeneity and sample imbalance: A transformer-based approach for battery degradation prediction in electric vehicles. Sensors, 2025, 25(11): 3564.
[24] Mienye ID, Swart TG, Obaido G. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. Information, 2024, 15(9): 517.
[25] Chen S, Liu Y, Zhang Q, et al. Multi-distance spatial-temporal graph neural network for anomaly detection in blockchain transactions. Advanced Intelligent Systems, 2025: 2400898.
[26] Fang Z. Adaptive QoS - Aware Cloud−Edge Collaborative Architecture for Real - Time Smart Water Service Management, 2025.
[27] Vajda DL, Do TV, Bérczes T, et al. Machine learning-based real-time anomaly detection using data pre-processing in the telemetry of server farms. Scientific Reports, 2024, 14(1): 23288.
[28] Shao Z, Wang X, Ji E, et al. GNN-EADD: Graph neural network-based e-commerce anomaly detection via dual-stage learning. IEEE Access, 2025.
[29] Gortney ME, Harris PE, Cerny T, et al. Visualizing microservice architecture in the dynamic perspective: A systematic mapping study. IEEE Access, 2022, 10: 119999–120012.
[30] Li P, Ren S, Zhang Q, et al. Think4SCND: Reinforcement learning with thinking model for dynamic supply chain network design. IEEE Access, 2024.
[31] Wang J, Tan Y, Jiang B, et al. Dynamic marketing uplift modeling: A symmetry-preserving framework integrating causal forests with deep reinforcement learning for personalized intervention strategies. Symmetry, 2025, 17(4): 610.
[32] Johnson R. Designing secure and scalable IoT systems: Definitive reference for developers and engineers. HiTeX Press, 2025.