

Anomaly Detection in Azure Cloud Infrastructure Using Machine Learning

Shekar Rao Lakavath

Independent Researcher, Bonneylake, Washington, USA

✉ **Corresponding Author:** Shekar Rao Lakavath, **E-mail:** Shekar.lakavath@gmail.com

ARTICLE INFORMATION

Submitted: June 20, 2026

Accepted: August 11, 2026

Published: October 03, 2026

Volume: 2

Issue: 1

KEYWORDS

Anomaly detection, machine learning, Microsoft Azure, cloud infrastructure, time-series analysis, Azure Monitor, log analysis

ABSTRACT

Cloud infrastructure generates a continuous, high-volume stream of telemetry, spanning resource utilization metrics, application performance traces, network flow data, and system logs, within which genuine failures and security incidents are rare events buried in a much larger volume of normal variation. Rule-based thresholding, the traditional approach to detecting infrastructure problems, struggles to keep pace with this volume and with workloads whose normal behavior shifts seasonally or evolves gradually over time. Machine learning offers a more adaptive alternative, learning the statistical or temporal structure of normal infrastructure behavior and flagging deviations without requiring every failure mode to be anticipated in advance. This article examines how machine learning techniques can be applied to detect anomalies in Microsoft Azure cloud infrastructure. The article proposes a machine learning pipeline spanning telemetry collection through Azure Monitor and Log Analytics, feature engineering, model-based anomaly scoring, and automated alerting through Azure Monitor and Sentinel, compares six machine learning techniques against the Azure infrastructure signals each is best suited to, and maps common categories of infrastructure anomaly to specific recommended techniques. Two tables and one pipeline architecture figure illustrate this analysis. The findings indicate that no single technique addresses every anomaly category; isolation-based methods suit high-dimensional resource metrics, autoencoder and variational approaches suit temporal and seasonal patterns, and transformer-based models suit unstructured log data, meaning a production Azure anomaly detection system typically requires a combination of techniques rather than a single model. The article concludes with practical recommendations for implementing machine-learning-based anomaly detection in Azure environments.

1. Introduction

Modern cloud infrastructure produces telemetry at a volume and velocity that exceeds what manual monitoring or static, rule-based thresholds can reliably interpret. A single Azure deployment may generate resource utilization metrics, application performance traces, network flow records, and system logs across hundreds or thousands of virtual machines, containers, and managed services simultaneously. Within this telemetry, the events that matter most, namely genuine failures, performance regressions, and security incidents, are comparatively rare and often resemble normal variation closely enough that fixed thresholds either miss them or generate an unsustainable volume of false alerts (Doelitzscher, Knahl, Reich, & Clarke, 2013).

Machine learning approaches to anomaly detection address this problem by learning the statistical or temporal structure of normal infrastructure behavior directly from historical telemetry, rather than requiring engineers to specify every failure condition as an explicit rule in advance. Early cloud-specific work in this area, such as Twitter's statistical decomposition approach to anomaly detection in cloud time series (Hochenbaum, Vallis, & Kejariwal, 2017), demonstrated that seasonal and trend decomposition combined with robust statistical outlier detection could substantially reduce false-positive rates relative to fixed thresholds. Subsequent research has extended this foundation with deep learning techniques capable of capturing more complex temporal and relational structure in infrastructure telemetry (Xu et al., 2018; Zuo, Wu, Min, Huang, & Pei, 2020).

Microsoft Azure provides the telemetry infrastructure, through Azure Monitor, Log Analytics, and Application Insights, needed to feed machine learning anomaly detection pipelines, and increasingly incorporates machine-learning-based detection directly into its own monitoring and security services. However, translating the broader machine learning anomaly detection literature into an effective Azure-specific implementation requires understanding which techniques suit which categories of infrastructure telemetry. This article addresses three questions. First, what machine learning techniques have been established in the literature for detecting anomalies in cloud infrastructure telemetry, and what are their respective strengths? Second, how can these techniques be organized into a coherent detection pipeline for Azure-hosted infrastructure? Third, which technique is best suited to which category of infrastructure anomaly? The remainder of the article reviews the relevant literature, proposes a machine learning anomaly detection pipeline for Azure, compares techniques against the infrastructure signals they suit best, maps anomaly categories to recommended techniques, and discusses practical implications.

2. Literature Review

2.1 Statistical Foundations of Cloud Anomaly Detection

Early cloud-specific anomaly detection research established statistical decomposition and robust outlier detection as a practical alternative to fixed thresholds. Doelitzscher, Knahl, Reich, and Clarke (2013) examined anomaly detection specifically within Infrastructure-as-a-Service clouds, finding that the dynamic, elastic nature of cloud resource allocation made static baselines unreliable and motivating detection approaches that adapt to changing normal behavior. Hochenbaum, Vallis, and Kejariwal (2017) proposed a statistical learning approach, later released as Twitter's widely adopted AnomalyDetection package, that decomposes time series into seasonal, trend, and residual components before applying a robust statistical test to the residual, substantially reducing false positives relative to naive thresholding on raw metric values.

Ahmad, Lavin, Purdy, and Agha (2017) proposed an unsupervised, online approach to real-time anomaly detection for streaming data using hierarchical temporal memory, a continuously learning sequence-memory algorithm capable of adapting to gradual concept drift in normal behavior without requiring periodic retraining, a property particularly relevant to cloud infrastructure whose normal operating characteristics shift as workloads and deployed services evolve.

2.2 Deep Learning for Time-Series Anomaly Detection

Deep learning approaches have extended anomaly detection to capture more complex temporal dependencies than statistical decomposition alone. Xu et al. (2018) proposed Donut, a variational autoencoder trained specifically to model the probability distribution of seasonal key performance indicators in web applications, demonstrating strong performance on the kind of periodic traffic and latency metrics common in cloud service monitoring. Amarbayasgalan, Pham, Theera-Umpon, and Ryu (2020) proposed an unsupervised deep reconstruction error approach for multi-domain time series anomaly detection, extending the autoencoder-based reconstruction paradigm across heterogeneous data domains without requiring domain-specific retraining for each new deployment.

Chen et al. (2021) proposed DAEMON, an unsupervised approach for detecting and interpreting anomalies in multivariate time series, addressing the practical need for detection systems to not only flag an anomaly but also indicate which specific metrics contributed to the anomalous score, a capability important for operational triage in cloud environments where a single anomaly alert may span dozens of correlated metrics. Shen, Yu, Ma, and Kwok (2021) proposed a multiresolution ensemble decoding approach that analyzes time series at multiple temporal granularities simultaneously, improving detection of anomalies that manifest differently depending on the time scale at which telemetry is aggregated. Borghesi, Bartolini, Lombardi, Milano, and Benini (2019) applied autoencoder-based anomaly detection specifically to high-performance computing systems, demonstrating that reconstruction-error-based detection generalizes effectively to the large-scale, resource-intensive workloads increasingly common in cloud-hosted machine learning and scientific computing infrastructure.

2.3 Cloud- and Microservice-Specific Anomaly Detection

As cloud applications have shifted toward microservice architectures, researchers have examined anomaly detection specifically within this distributed context. Islam and Miranskyy (2020) proposed an anomaly detection approach specifically for cloud components, and Islam, Pourmajidi, Zhang, Steinbacher, Erwin, and Miranskyy (2021) extended this work with a large-scale industrial case study of anomaly detection deployed across a production cloud platform, finding that combining multiple detection signals, rather than relying on any single metric, substantially improved detection precision in a real-world operational environment. Zuo, Wu, Min, Huang, and Pei (2020) proposed an anomaly detection scheme specifically designed for microservice architectures that jointly analyzes temporal and spatial (inter-service dependency) data, addressing the challenge that a single microservice's anomalous behavior often manifests as a cascading pattern across dependent services rather than an isolated event.

Kohyarnjadfard, Aloise, Azhari, and Dagenais (2022) proposed an anomaly detection approach for microservice environments that combines distributed tracing data with natural language processing techniques, allowing the system to identify anomalies described in unstructured trace metadata that purely numerical metric analysis would miss. Soldani and Brogi (2022) conducted a comprehensive survey of anomaly detection and failure root cause analysis specifically for microservice-based cloud applications, concluding that root cause analysis, not merely anomaly detection, represents the more operationally valuable but substantially less mature research direction, since identifying that an anomaly occurred is less actionable than identifying which specific service or dependency caused it.

2.4 Log-Based Anomaly Detection

System and application logs represent a distinct and unstructured category of cloud telemetry requiring specialized detection techniques. Miranskyy, Hamou-Lhadj, Cialini, and Larsson (2016) examined the challenges of operational log analysis at big-data scale, identifying log volume, format heterogeneity, and the absence of labeled anomaly examples as the primary obstacles to effective automated log analysis in production cloud systems. Guo, Yuan, and Wu (2021) proposed LogBERT, a transformer-based language model adapted specifically for log anomaly detection, demonstrating that pretrained language modeling techniques originally developed for natural language can be effectively fine-tuned to detect semantically abnormal log sequences without requiring hand-engineered log parsing rules.

Elsayed, Le-Khac, Dev, and Jurcut (2020) proposed a network anomaly detection approach using an LSTM-based autoencoder, extending log- and metric-based anomaly detection specifically to network traffic telemetry, an increasingly important signal category as cloud infrastructure security monitoring converges with conventional network intrusion detection.

2.5 Surveys and Recent Advances

Several comprehensive surveys have synthesized the anomaly detection literature specifically for cloud and high-dimensional data contexts. Thudumu, Branch, Jin, and Singh (2020) surveyed anomaly detection techniques for high-dimensional big data broadly, concluding that dimensionality reduction remains a critical preprocessing step for most machine learning detection techniques applied to the wide, correlated metric sets typical of cloud infrastructure monitoring. Hagemann and Katsarou (2020) conducted a systematic review specifically of anomaly detection for cloud computing environments, finding that despite substantial research activity, comparatively few studies evaluated their proposed techniques against production-scale, real-world cloud telemetry rather than synthetic or laboratory benchmarks.

Al-amri, Murugesan, Man, Abdulateef, Al-Sharafi, and Alkahtani (2021) reviewed machine learning and deep learning techniques for anomaly detection specifically in Internet of Things data, a telemetry category increasingly relevant to cloud infrastructure as connected devices generate data ingested into cloud-hosted monitoring pipelines. Chatterjee and Ahmed (2022) extended this review with a broader survey of Internet of Things anomaly detection methods and applications, identifying resource-constrained on-device inference as a persistent tension with the more computationally intensive deep learning models that achieve the best detection accuracy in cloud-hosted evaluation.

More recent work has continued to refine cloud-specific anomaly detection. Zhong et al. (2024) proposed a broad network-based contrastive autoencoder for detecting cloud anomalies, demonstrating improved performance over conventional reconstruction-based autoencoders by explicitly contrasting normal and anomalous representations during training. Abdallah et al. (2024) conducted a recent comprehensive review of machine and deep learning techniques for cloud network anomaly detection, concluding that hybrid approaches combining statistical, machine learning, and deep learning techniques consistently outperformed any single technique category across the benchmark studies they reviewed. Almazrawe and Musawi (2024) provided a further recent review specifically focused on anomaly detection in cloud networks, reinforcing the conclusion that no single algorithmic approach is uniformly superior across all cloud anomaly detection tasks.

Taken together, this literature indicates that effective anomaly detection in cloud infrastructure requires matching the detection technique to the specific characteristics of the telemetry signal: statistical decomposition and isolation-based methods for high-dimensional resource metrics, autoencoder and variational approaches for temporal and seasonal patterns, graph-based methods for interdependent microservice telemetry, and transformer-based models for unstructured log data. The following sections apply this understanding specifically to Microsoft Azure infrastructure.

3. A Machine Learning Pipeline for Anomaly Detection in Azure Infrastructure

Building on the multi-technique detection approach validated across the literature reviewed above (Islam, Pourmajidi, Zhang, Steinbacher, Erwin, & Miranskyy, 2021; Abdallah et al., 2024), this article proposes a machine learning pipeline for detecting anomalies in Azure cloud infrastructure. Figure 1 illustrates the pipeline, spanning telemetry collection, feature engineering, model-based detection, anomaly scoring, and automated alerting, with an analyst feedback loop that continuously refines detection models and thresholds.

Figure 1. Machine Learning Pipeline for Anomaly Detection in Azure Cloud Infrastructure

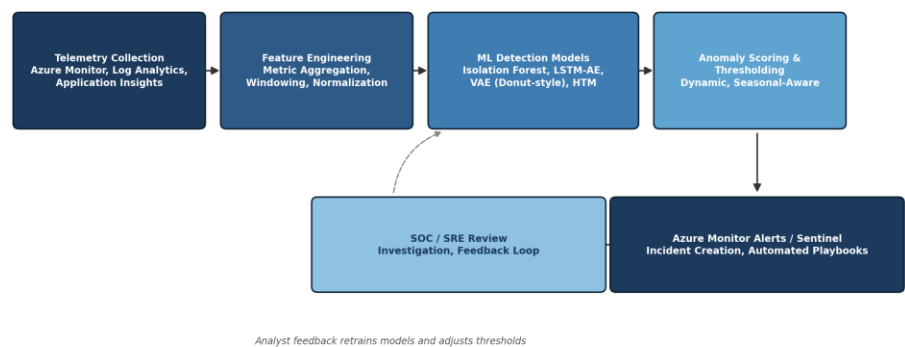


Figure 1. Machine learning pipeline for anomaly detection in Azure cloud infrastructure, showing the flow from telemetry collection through feature engineering, ML detection models, anomaly scoring, and automated alerting, with a feedback loop from analyst review back into model retraining

3.1 Telemetry Collection

Azure Monitor, Log Analytics, and Application Insights collect resource metrics, distributed traces, and application logs across the monitored environment, providing the raw telemetry substrate that Miranskyy, Hamou-Lhadj, Cialini, and Larsson (2016) identify as the necessary but insufficient starting point for effective log and metric analysis at scale.

3.2 Feature Engineering

Raw telemetry is aggregated into fixed-size windows, normalized, and, where relevant, decomposed into seasonal, trend, and residual components following the approach validated by Hochenbaum, Vallis, and Kejariwal (2017), preparing the data for the specific requirements of each downstream detection model.

3.3 ML Detection Models

Multiple detection models operate in parallel, each suited to a different telemetry characteristic: isolation-based methods for high-dimensional resource metrics, LSTM autoencoders and variational autoencoders for temporal and seasonal patterns (Xu et al., 2018), and hierarchical temporal memory for streaming telemetry subject to gradual concept drift (Ahmad, Lavin, Purdy, & Agha, 2017). This multi-model design reflects the hybrid-approach conclusion reached by Abdallah et al. (2024) in their review of cloud network anomaly detection techniques.

3.4 Anomaly Scoring, Alerting, and Feedback

Detection outputs are combined into a unified anomaly score with dynamic, seasonally aware thresholds, and scores exceeding threshold trigger incident creation in Azure Monitor Alerts or Microsoft Sentinel, consistent with the multi-signal precision improvement documented by Islam, Pourmajidi, Zhang, Steinbacher, Erwin, and Miranskyy (2021). Analyst investigation outcomes feed back into model retraining and threshold adjustment, closing the loop that Soldani and Brogi (2022) identify as essential for the detection system to remain effective as infrastructure and workloads evolve.

4. Comparing Machine Learning Techniques for Azure Infrastructure Signals

Table 1 compares six machine learning techniques established in the literature against the specific category of Azure infrastructure telemetry each is best suited to detect anomalies within.

Table 1. Machine Learning Techniques Compared Against Azure Infrastructure Signal Types

ML Technique	Underlying Approach	Best Suited Azure Infrastructure Signal
Isolation Forest	Isolates observations via random recursive partitioning; anomalies require fewer splits to isolate	High-dimensional resource metrics (CPU, memory, disk I/O) with limited seasonality

LSTM Autoencoder	Learns to reconstruct normal temporal sequences; high reconstruction error signals anomalies	Sequential performance metrics with temporal dependencies, such as request latency over time
Variational Autoencoder (Donut-style)	Models the probability distribution of seasonal KPI data; low likelihood indicates anomaly	Seasonal, periodic workload metrics such as daily or weekly traffic patterns
Hierarchical Temporal Memory (HTM)	Online, continuously learning sequence memory that adapts to concept drift without retraining	Streaming telemetry where normal behavior evolves gradually over time
Graph Attention Networks	Models dependencies between interacting components to detect anomalies in relational structure	Multivariate telemetry across interdependent microservices
BERT-based Log Models	Applies transformer-based language modeling to log sequences to flag semantically abnormal entries	Unstructured application and system log streams

As Table 1 indicates, no single technique is universally applicable; isolation-based methods excel at high-dimensional, less seasonal resource metrics, while variational and hierarchical temporal memory approaches are specifically suited to the periodic and gradually evolving patterns common in cloud workload telemetry, consistent with the hybrid-detection conclusion reached by Abdallah et al. (2024).

5. Mapping Anomaly Types to Recommended Techniques

Table 2 synthesizes the categories of anomaly most frequently documented in the cloud infrastructure literature reviewed above and maps each to a recommended machine learning technique, along with representative supporting sources.

Table 2. Azure Infrastructure Anomaly Types Mapped to Recommended ML Techniques

Anomaly Type in Azure Infrastructure	Recommended ML Technique	Supporting Literature
Sudden resource spikes (CPU, memory, disk) without a corresponding workload increase	Isolation Forest or statistical control-chart methods on Azure Monitor metrics	Hochenbaum, Vallis, & Kejariwal (2017); Doelitzscher, Knahl, Reich, & Clarke (2013)
Gradual performance degradation across a distributed microservice call chain	Graph attention or dependency-aware anomaly detection across Application Insights traces	Zuo, Wu, Min, Huang, & Pei (2020); Kohyarnejadfar, Aloise, Azhari, & Dagenais (2022)
Seasonal traffic pattern deviations (unexpected dips or surges)	Variational autoencoder trained on historical seasonal KPI baselines	Xu et al. (2018); Qiu, Du, & Qian (2019)
Anomalous log entries indicating misconfiguration or failure	Transformer-based log anomaly detection over Log Analytics log streams	Guo, Yuan, & Wu (2021); Miranskyy, Hamou-Lhadj, Cialini, & Larsson (2016)
Network-layer anomalies (unusual traffic volume or flow)	LSTM-based autoencoder over network flow telemetry	Elsayed, Le-Khac, Dev, & Jurcut (2020); Abdallah et al. (2024)

patterns)		
Slow concept drift in normal system behavior over weeks or months	Online, continuously adapting models such as hierarchical temporal memory	Ahmad, Lavin, Purdy, & Agha (2017)
Rare, high-dimensional multivariate anomalies across correlated metrics	Deep autoencoding or ensemble decoding approaches capturing cross-metric correlation	Chen et al. (2021); Shen, Yu, Ma, & Kwok (2021)

The mapping in Table 2 reinforces that a production Azure anomaly detection system should be organized around anomaly category rather than a single universal model; a system tuned only for sudden resource spikes, for example, is unlikely to detect the gradual, cascading performance degradation patterns that Zuo, Wu, Min, Huang, and Pei (2020) identify as characteristic of microservice dependency failures.

6. Discussion

The analysis presented above supports several practical conclusions for teams implementing machine-learning-based anomaly detection on Azure. First, seasonal and trend decomposition should precede model-based detection wherever workload telemetry exhibits periodic patterns; Hochenbaum, Vallis, and Kejariwal's (2017) statistical decomposition approach and Xu et al.'s (2018) variational autoencoder both demonstrate that accounting for seasonality substantially reduces false positives relative to applying anomaly detection directly to raw, undecomposed metric values.

Second, detection technique selection should be driven by the specific characteristics of the telemetry signal rather than a single default model applied uniformly across all metrics. The comparative analysis in Table 1 indicates that isolation-based methods, autoencoder variants, and transformer-based log models each address a distinct category of telemetry, and Abdallah et al.'s (2024) finding that hybrid approaches outperform any single technique category reinforces that production systems should combine multiple detection models rather than standardizing on one.

Third, detecting an anomaly is necessary but not sufficient for operational value; Soldani and Brogi (2022) identify root cause analysis as the more operationally valuable but less mature capability, meaning organizations should evaluate anomaly detection systems partly by how effectively they support subsequent triage, for example through the interpretable anomaly attribution that Chen et al. (2021) demonstrate with DAEMON, rather than by detection accuracy alone.

Fourth, microservice architectures require detection approaches that account for inter-service dependency rather than treating each service's telemetry in isolation. Zuo, Wu, Min, Huang, and Pei's (2020) joint temporal-spatial analysis and Kohyarnjadfard, Aloise, Azhari, and Dagenais's (2022) trace-based approach both demonstrate that anomalies in distributed applications frequently manifest as cascading patterns across dependent services, a pattern that per-service threshold monitoring is structurally unable to detect.

Finally, log data requires fundamentally different modeling techniques than numerical metrics. Miranskyy, Hamou-Lhadj, Cialini, and Larsson's (2016) identification of format heterogeneity as a persistent obstacle, combined with Guo, Yuan, and Wu's (2021) demonstration that transformer-based language models can be effectively adapted to log anomaly detection, together suggest that organizations should treat log anomaly detection as a distinct pipeline component rather than attempting to force unstructured log data into the same numerical anomaly detection models applied to resource metrics.

7. Recommendations

Organizations implementing machine-learning-based anomaly detection in Azure infrastructure should: (1) apply seasonal and trend decomposition to workload telemetry before model-based detection, following the approach validated by Hochenbaum, Vallis, and Kejariwal (2017); (2) deploy multiple detection techniques matched to telemetry type rather than a single universal model, combining isolation-based methods for resource metrics, autoencoder variants for temporal and seasonal patterns, and transformer-based models for log data; (3) incorporate inter-service dependency information into detection logic for microservice architectures, following Zuo, Wu, Min, Huang, and Pei (2020), rather than monitoring each service independently; (4) prioritize interpretable, attribution-capable detection models such as DAEMON (Chen et al., 2021) that indicate which specific metrics or services contributed to an anomaly, supporting faster analyst triage; (5) establish a continuous feedback loop in which analyst investigation outcomes retrain detection models and adjust thresholds, consistent with the operational precision

improvements documented by Islam, Pourmajidi, Zhang, Steinbacher, Erwin, and Miranskyy (2021); and (6) evaluate detection systems against production-representative telemetry rather than synthetic benchmarks alone, addressing the evaluation gap that Hagemann and Katsarou (2020) identify as common across the existing literature.

8. Conclusion

Machine learning has become an essential complement to traditional rule-based monitoring for detecting anomalies in the high-volume, high-dimensional telemetry that modern Azure cloud infrastructure generates. This article has reviewed the evolution of anomaly detection techniques from early statistical decomposition methods through deep-learning-based autoencoders, graph-based microservice detection, and transformer-based log analysis, proposed a machine learning pipeline for Azure infrastructure spanning telemetry collection through automated alerting, and mapped specific anomaly categories to the techniques best suited to detect them. The literature reviewed indicates that no single machine learning technique addresses every category of infrastructure anomaly; effective production systems combine multiple detection approaches, each matched to the statistical and temporal characteristics of a specific telemetry signal, and incorporate a continuous feedback loop that allows detection models to adapt as infrastructure and workloads evolve. Future research should evaluate these techniques against larger, longitudinal production Azure telemetry datasets to more precisely quantify the detection accuracy and operational overhead tradeoffs identified in this review.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers

Artificial Intelligence (AI) Use Disclosure: The authors declare that no artificial intelligence tools were used in the preparation of this manuscript.

References

- [1] Abdallah, A. M., Alkaabi, A. S. R. O., Alameri, G. B. N. D., Rafique, S. H., Musa, N. S., & Murugan, T. (2024). Cloud network anomaly detection using machine and deep learning techniques—recent research advancements. *IEEE Access*, 12, 56749–56773. <https://doi.org/10.1109/ACCESS.2024.3390844>
- [2] Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262, 134–147. <https://doi.org/10.1016/j.neucom.2017.04.070>
- [3] Al-amri, R., Murugesan, R. K., Man, M., Abdulateef, A. F., Al-Sharafi, M. A., & Alkahtani, A. A. (2021). A review of machine learning and deep learning techniques for anomaly detection in IoT data. *Applied Sciences*, 11(12), 5320. <https://doi.org/10.3390/app11125320>
- [4] Almazrawe, A., & Musawi, B. (2024). Anomaly detection in cloud network: A review. *BIO Web of Conferences*, 97, 00019. <https://doi.org/10.1051/bioconf/20249700019>
- [5] Amarbayasgalan, T., Pham, V. H., Theera-Umpon, N., & Ryu, K. H. (2020). Unsupervised anomaly detection approach for time-series in multi-domains using deep reconstruction error. *Symmetry*, 12(8), 1251. <https://doi.org/10.3390/sym12081251>
- [6] Borghesi, A., Bartolini, M., Lombardi, M., Milano, M., & Benini, L. (2019). Anomaly detection using autoencoders in high performance computing systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1), 9428–9433. <https://doi.org/10.1609/aaai.v33i01.33019428>
- [7] Chatterjee, A., & Ahmed, B. S. (2022). IoT anomaly detection methods and applications: A survey. *Internet of Things*, 19, 100568. <https://doi.org/10.1016/j.iot.2022.100568>
- [8] Chen, X., Deng, L., Huang, F., Zhang, C., Zhang, Z., Zhao, Y., & Zheng, K. (2021). DAEMON: Unsupervised anomaly detection and interpretation for multivariate time series. In 2021 IEEE 37th International Conference on Data Engineering (ICDE) (pp. 2225–2230). IEEE. <https://doi.org/10.1109/ICDE51399.2021.00219>
- [9] Doelitzscher, F., Knahl, M., Reich, C., & Clarke, N. (2013). Anomaly detection in IaaS clouds. In 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (Vol. 1, pp. 387–394). IEEE. <https://doi.org/10.1109/CloudCom.2013.57>
- [10] Elsayed, M. S., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2020). Network anomaly detection using LSTM based autoencoder. In Q2SWinet '20: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks (pp. 37–45). ACM. <https://doi.org/10.1145/3416013.3426457>
- [11] Guo, H., Yuan, S., & Wu, X. (2021). LogBERT: Log anomaly detection via BERT. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1–8). IEEE. <https://doi.org/10.1109/IJCNN52387.2021.9534113>
- [12] Hagemann, T., & Katsarou, K. (2020). A systematic review on anomaly detection for cloud computing environments. In Proceedings of the 2020 3rd Artificial Intelligence and Cloud Computing Conference (pp. 83–96). ACM. <https://doi.org/10.1145/3442536.3442550>
- [13] Hochenbaum, J., Vallis, O. S., & Kejariwal, A. (2017). Automatic anomaly detection in the cloud via statistical learning (arXiv:1704.07706). arXiv. <https://doi.org/10.48550/arXiv.1704.07706>
- [14] Islam, M. S., & Miranskyy, A. (2020). Anomaly detection in cloud components. In 2020 IEEE 13th International Conference on Cloud Computing (CLOUD) (pp. 1–3). IEEE. <https://doi.org/10.1109/CLOUD49709.2020.00008>
- [15] Islam, M. S., Pourmajidi, W., Zhang, L., Steinbacher, J., Erwin, T., & Miranskyy, A. (2021). Anomaly detection in a large-scale cloud platform. In 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 150–159). IEEE. <https://doi.org/10.1109/ICSE-SEIP52600.2021.00024>

- [16] Kohyarnjadfard, I., Aloise, D., Azhari, S. V., & Dagenais, M. R. (2022). Anomaly detection in microservice environments using distributed tracing data and NLP. *Journal of Cloud Computing*, 11, 25. <https://doi.org/10.1186/s13677-022-00296-4>
- [17] Miransky, A., Hamou-Lhadj, A., Cialini, E., & Larsson, A. (2016). Operational-log analysis for big data systems: Challenges and solutions. *IEEE Software*, 33(2), 52–59. <https://doi.org/10.1109/MS.2016.33>
- [18] Ram Shankar Siva Kumar, Wicker, A., & Swann, M. (2017). Practical machine learning for cloud intrusion detection: Challenges and the way forward. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (pp. 81–90). ACM. <https://doi.org/10.1145/3128572.3140445>
- [19] Qiu, J., Du, Q., & Qian, C. (2019). KPI-TSAD: A time-series anomaly detector for KPI monitoring in cloud applications. *Symmetry*, 11(11), 1350. <https://doi.org/10.3390/sym11111350>
- [20] Shen, L., Yu, Z., Ma, Q., & Kwok, J. T. (2021). Time series anomaly detection with multiresolution ensemble decoding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11), 9567–9575. <https://doi.org/10.1609/aaai.v35i11.17152>
- [21] Soldani, J., & Brogi, A. (2022). Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey. *ACM Computing Surveys*, 55(3), 1–39. <https://doi.org/10.1145/3501297>
- [22] Thudumu, S., Branch, P., Jin, J., & Singh, J. (2020). A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7, 42. <https://doi.org/10.1186/s40537-020-00320-x>
- [23] Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., Chen, J., Wang, Z., & Qiao, H. (2018). Unsupervised anomaly detection via variational autoencoder for seasonal KPIs in web applications. In *Proceedings of the 2018 World Wide Web Conference* (pp. 187–196). ACM. <https://doi.org/10.1145/3178876.3185996>
- [24] Zhong, G., Liu, F., et al. (2024). Detecting cloud anomaly via broad network-based contrastive autoencoder. *IEEE Transactions on Network and Service Management*, 21(3), 3249–3263. <https://doi.org/10.1109/TNSM.2024.3364381>
- [25] Zuo, Y., Wu, Y., Min, G., Huang, C., & Pei, K. (2020). An intelligent anomaly detection scheme for micro-services architectures with temporal and spatial data analysis. *IEEE Transactions on Cognitive Communications and Networking*, 6(2), 548–561. <https://doi.org/10.1109/TCCN.2020.2966615>