International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group | Volume 7 Issue 2 PP 19-26, 2026
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V7I2P103

Original Article

AI-Based Anomaly Detection for Cloud Logs and Distributed Traces

Srija Rajak
Independent Researcher, University of the Cumberlands.

Received On: 19/02/2026 ***Revised On: 22/03/2026*** ***Accepted On: 28/03/2026*** ***Published On: 05/04/2026***

Abstract *- Cloud-native infrastructures built on microservices, containers, and distributed tracing systems generate massive volumes of heterogeneous telemetry data, including logs, metrics, and traces. Traditional rule-based anomaly detection techniques are increasingly inadequate due to their rigidity, high false positive rates, and inability to adapt to dynamic workloads. This paper introduces a machine learning-based anomaly detection model of cloud logs and distributed traces that uses machine learning and deep learning frameworks to detect anomalous behavior in a system in real-time. The proposed system incorporates multi-modal telemetry over a scalable Kubernetes implementation and implements methods including Isolation Forest and Autoencoders to perform unsupervised anomaly detection. By fusing log and trace data, the framework captures both semantic and structural dependencies across distributed services. Experimental evaluation in a simulated cloud environment demonstrates strong performance, achieving precision of up to 0.92, recall of 0.95, and F1-score of 0.94 for container escape detection, while maintaining an AUC above 0.95 across scenarios. Resource abuse detection had a precision of 0.91 and recall of 0.93, and unauthorized API usage had an F1-score of 0.89. These outcomes demonstrate high accuracy, memory and strength over the baseline models, which is why the method can be used in contemporary cloud security applications. It also addresses the aspect of scaling and real-time deployment in the setting of a production environment.*

Keywords *- Cloud Security, Kubernetes, Artificial Intelligence, Anomaly Detection, Log Analysis, Distributed Tracing.*

1. Introduction

Microservice-based, container-based, and orchestration-based (e.g., Kubernetes) cloud-native systems have revolutionized modern application deployment and management [1][2]. These are very distributed, dynamic and scalable environments that facilitate quick development and continuous service delivery [3][4][5]. Nevertheless, this change of architecture has also brought a lot of complexity to monitoring and security management of the systems [6][7]. Due to the decomposition of applications into multiple services, they produce huge amounts of heterogeneous observability data, such as logs, metrics, and distributed traces [8]. These telemetry sources are the most important sources of information on the behavior of the system, its performance, and security events, but due to their size and variety, they cannot be analyzed manually and require a more conventional monitoring strategy [9].

Traditional anomaly detection methods, such as rule-based systems, threshold-based monitoring, and statistical techniques [10], have difficulties in adjusting to the dynamically changing workloads and changing patterns of attacks in the cloud. These techniques tend to be inflexible, need a lot of manual setup, and produce large false-positive rates, lowering their usefulness in practical applications [11][12]. In addition, they do not provide complex temporal and structural dependencies that exist in distributed systems. To overcome these shortcomings, there has been a surge towards smart and adaptive anomaly detection methods that can learn normal behavior patterns based on data that do not depend on preset rules.

The latest developments in artificial intelligence and machine learning have facilitated more solid and scalable solutions to cloud security [13][14]. Architectures based on deep learning have shown powerful features to recognize anomalies in high-dimensional and noisy telemetry data [15]. Specifically, hierarchical features can be automatically learned with deep learning models, which can learn both long-term and short-term dependencies in system behavior [16]. Moreover, logs, metrics and traces can be multimodally fused to gain a deeper insight into system interactions which leads to better detection and lesser blind spots in monitoring.

Although these are advanced, there are still challenges that are still unsolved. Cloud environments are non-stationary and cause concept drift in models, resulting in performance changes over time [17][18]. Moreover, deep learning models are currently expensive to compute, making them difficult to implement on a large scale, in real-time. The issue of interpretability of AI-based decisions is also a serious one and particularly when it comes to applications where security is a concern [19]. This study intends to solve these difficulties by presenting an AI-based anomaly detection platform that incorporates multimodal telemetry data inside a scalable architecture. The framework takes advantage of machine learning and deep learning to provide real-time, adaptive, and accurate detection of anomalies in cloud-native environments to enhance the reliability and security of the system.

1.1. Motivation and Contributions of the paper

Microservices, containers, and serverless computing are now widely used, leading to an increase in the distribution, dynamism, and sensitivity of a modern cloud environment to security. This complexity gives rise to huge quantities of logs, traces, and metrics that cannot be analyzed with traditional rule-based monitoring systems. Unauthorized access, resource abuse, and container escape attacks are some of the security threats that are constantly evolving and demand intelligent and adaptive detection mechanisms. This work is motivated by the need to create an AI-based anomaly detection model that will learn normal behavior using multimodal telemetry data and detect anomalies in real time to improve security. The major contributions of the paper are:

- Proposes an AI-based anomaly detection framework for cloud logs and distributed traces using multimodal telemetry fusion.
- Integrates logs, metrics, and traces within a unified Kubernetes-based scalable architecture.
- Uses autoencoders and isolation forests, two unsupervised learning methods, to identify anomalies.
- Enables real-time detection of security threats including resource abuse, API misuse, and container escape attempts.
- Provides a comparative evaluation against baseline models demonstrating improved prec, rec, and F1.

1.2. Justification and Novelty of the paper

This study is justified by the increasing complexity of cloud-native infrastructures, where traditional monitoring systems fail to detect sophisticated and dynamic security threats. The originality of the suggested solution is its multimodal combination of logs, traces, and metrics in a single AI-based data anomaly detection system. In contrast to traditional approach, which uses isolated sources of data or fixed thresholds, the proposed system utilizes unsupervised deep learning and machine learning algorithms to extract both time-related and structural dependencies of distributed systems. This makes it possible to detect anomalies in clouds more accurately, scalably and adaptively.

1.3. Organization of the paper

This paper will be divided into sections to offer the proposed work. Section II presents a literature overview on several learning approaches, including conventional, ML and DL. Section III outlines the approach and design of the proposed system. Results and performance evaluation are covered in Section IV. The work is concluded and future research directions are outlined in Section V.

2. Background and Related Work

Cloud computing anomaly detection is a key research area due to the widespread adoption of cloud-native, microservice, and distributed systems that generate large-scale observability data. Although cloud platforms provide scalability and flexibility, they also introduce complexity and dynamic behavior that challenge traditional monitoring methods. Rule-based, threshold-based, and statistical approaches are often unable to adapt to evolving workloads. In contrast, ML and DL techniques are more effective for handling high-dimensional and complex cloud observability data.

2.1. Historical Perspectives and Traditional Methods

Conventional methods of anomaly detection in distributed systems typically rely on statistical techniques such as control charts, thresholding and clustering to identify outliers in telemetry data streams. E. Malul et al. (2024) present GenKubeSec, an adaptive and all-encompassing LLM-based approach that not only finds many different types of KCF misconfigurations but also pinpoints their precise locations, gives thorough justifications for them, and suggests ways to fix them. Compared experimentally with three gold-standard RB tools in the business [20]. These methods assume an unknown distribution is an aberrant behavior, whereas non-stationary loads and intricate dependencies that are observed in cloud loads are troublesome to them. Thus, only the static methods are not sufficient in the environment of the modern cloud context where the work load profile and the dynamism of the performance undergo constant changes.

2.2. Machine Learning-Based Approaches

The development of ML techniques which enable the study of normal behavior patterns through both labeled and unlabelled data brought the research field a significant advancement. Supervised paradigms are used to predict anomalies based on the training of those instances, whereas unsupervised paradigms (e.g. clustering, density based approaches) are used to identify deviations by modeling the normal distribution of patterns M. S. Islam et al., (2025) present a newly generated high-dimensional dataset sourced from IBM Cloud, amassed over a period of four and a half months using the IBM Cloud Console. The telemetry data in this collection is organized into 39,365 rows and 117,448 columns. It makes it easier to test anomaly detection systems on actual data, which is a step in the right direction for creating reliable solutions to keep large-scale cloud infrastructures running well [21]. Although ML models are relatively effective in specific scenarios, they can fail with high-dimensional data, noisy data, and multimodal cloud observability data - which is the case with distributed systems telemetry on its own. Moreover, the feature-extraction bottleneck in classical ML often requires substantial domain expertise and may fail to learn complex interactions between modalities (e.g., log events associated with trace paths).

2.3. Deep Learning for Anomaly Detection

The deep learning architectures provide automated representation learning in which models do not require significant manual feature engineering to learn the complex nonlinear relationships. In order to effectively identify unusual data breaches in decentralized cloud storage facilities, S. Potluri (2024) suggests a dependable approach that employs deep learning. To sort through the deluge of log data and information generated by cloud technologies, the system employs a combination of Autoencoders, Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) networks. The proposed model achieved 98.7% detection accuracy and a 1.2% false positive rate, surpassing state-of-the-art methods. Additionally, the framework is

adaptable, scalable, and interoperable with a wide range of cloud providers [22]. Deep models are also more general to various log formats and noise properties by default through hierarchical feature extraction.

2.4. Log-Based Anomaly Detection

Log data are a first-order source of observability, and they contain useful data regarding the state of systems, their errors, and their behavior in distributed applications. Log-based anomaly detection focuses on identifying patterns of events that deviate from normal sequences. X. Wei et al., (2024) provide a distributed system a generic LAD foundation. Two essential parts of LAD that affect the effectiveness of anomaly detection are log grouping and feature pattern mining. This paper will give a comprehensive overview of methods in these two areas, as well as classification frameworks for log grouping and feature patterns, and will summarize four log-grouping techniques and five feature patterns. Present the results of applying these methods to Ray, a well-known industrial distributed system, in order to assess their usefulness [23]. A generic issue in cloud environments where the log format and frequency of events vary with time.

2.5. Trace-Based Anomaly Detection

Distributed tracing is a way to allow visibility into the flow of requests through services in microservice contexts on the path level. The Traces are a sequence of spans representing timing, context, causal dependencies among invocations to a service that is invaluable in either identifying or discovering the performance anomalies occurring between multiple components. Trace-based anomaly detection performs analysis on these multi-span structures and determines the odd invocation patterns or performance anomalies. C. Zhang et al. (2022) suggests using DeepTraLog, a method for detecting anomalies in microservices that is based on deep learning. The complex structure of a trace and the log events it contains can be represented by DeepTraLog as a single graph. Using the graph representation, DeepTraLog combines traces and logs to train a deep SVDD model using GGNNs. To identify anomalies in newly generated traces and the accompanying logs, it employs this paradigm. In a microservice benchmark, DeepTraLog achieves better results than state-of-the-art trace/log anomaly detection algorithms, with an average F1-score improvement of 0.37. It also shows good recall (0.97) and accuracy (0.93) [24]. These trace-based methods represent an example of the movement towards modeling the behavior of systems holistically, as opposed to the analysis of isolated signals.

2.6. Hybrid Log-Trace Approaches

Although logs and traces alone can provide useful information, hybrid approaches that use both modalities tend to yield better results, as they focus on the complementary behaviors of systems. C. Zhang et al. (2022) provide DeepTraLog, a deep learning-based approach to anomaly detection in microservices. While describing the complex structure of a trace and the log events. It achieves a recall of 0.93 and an accuracy of 0.97. DeepTraLog achieves an F1-score improvement of 0.37 over state-of-the-art trace/log anomaly detection algorithms on a microservice benchmark. Additional validations include DeepTraLog's efficiency, the unified graph representation's contribution, and the effects of setting up a few critical parameters [24].

Table I presents a comparison of cloud anomaly detection methods using classical, ML, DL, and hybrid methodologies, illuminating their respective strengths and weaknesses as well as research gaps that may be filled to provide more effective and scalable solutions.

Table 1: Comparative Analysis and Research Gaps in Cloud-Based Anomaly Detection Approaches

Category	Methods	Key Contributions	Limitations	Recommendations
Traditional -Based	Thresholding, statistical rules, GenKubeSec (LLM-based)	Detects misconfigurations with reasoning and remediation support	Limited adaptability to dynamic cloud environments; poor scalability for real-time anomaly detection	Develop adaptive, learning-based models that can dynamically adjust to changing workloads and system behaviors
Machine Learning	SVM, Random Forest, Clustering, Density-based methods	Provides real-world dataset; enables anomaly detection using structured telemetry	Struggles with high-dimensional, noisy, multimodal data; heavy feature engineering required	Use automated feature learning and dimensionality reduction techniques; integrate multimodal data sources
Deep Learning	CNN + LSTM + Autoencoder hybrid model	High accuracy (98.7%); captures temporal and spatial dependencies	High computational cost; low interpretability; limited multimodal integration	Design lightweight and interpretable DL models; incorporate explainable AI and multimodal fusion
Log-Based Detection	The LAD framework integrates log grouping with feature pattern mining	Systematic framework for log analysis; effective for distributed systems	Sensitive to log format variability; lacks system-level structural context	Combine log data with traces and metrics; improve robustness to log evolution

Trace-Based Detection	DeepTraLog (GGNN + Deep SVDD)	Captures service dependencies and execution flow; high precision and recall	Limited semantic depth without full log integration; may miss fine-grained event details	Enhance semantic enrichment by deeper log integration and temporal modeling
Hybrid Log-Trace	DeepTraLog (Graph-based fusion of logs and traces)	Combines structural and semantic information; improved anomaly detection performance	Limited scalability in large-scale real-time systems; insufficient temporal dynamics modeling	Develop scalable multimodal frameworks incorporating logs, traces, and metrics with temporal attention mechanisms

2.7. Emerging trends and Research gaps

Anomaly detection within cloud computing environments has made sizable progress yet remains unfinished because two critical research areas still require attention. The conventional techniques present us with a problem because they cannot adapt to changing cloud workloads that exhibit unpredictable behavior. The use of machine learning methods enables better detection results, but the systems require time-consuming manual work to develop operational systems because they struggle to analyze complex data from multiple sources. The deep learning models enable automatic feature extraction to overcome certain obstacles yet they create new issues through their need for excessive processing power and their inability to generate understandable results and their restricted capacity to operate in real time. The log-based and trace-based approaches produce incomplete results because they only examine either the semantic event information or the structural execution patterns. The log-trace integration hybrid method produces positive outcomes, but it still struggles to handle three core challenges, which include temporal modeling and full multimodal fusion and scalability. Therefore, there is a need for scalable, interpretable, and real-time frameworks that effectively integrate logs, traces, and metrics to enhance anomaly detection in complex cloud systems.

3. Methodology

The proposed system architecture for anomaly detection of cloud logs and distributed traces can support the scalability and complexity of contemporary cloud environments. It combines multiple sources of data between the control plane, service logs and runtime telemetry to create an entire pipeline capable of analyzing multi-modal data to identify possible anomalies. It is an architecture that integrates the best ML and high-performance data-processing systems to deliver real-time, self-adjusting anomaly detection in cloud-native environments.

3.1. Architectural Overview

There are four main parts to the system, which is designed to be modular and multi-layered: (i) Data Collection Module, (ii) Feature Engineering Layer, (iii) Anomaly Detection Layer, and (iv) Evaluation & Alerting Layer shown in Fig.1. The system uses messaging queues to enable its different layers to communicate with each other while achieving scalability and fault tolerance across all microservices, virtual machines and containerized environments.

The Data Collection Module collects various telemetry data, which it obtains by recording both Kubernetes API server audit logs that track user activity and system traces that show service operations. The system uses eBPF (extended Berkeley Packet Filter) probes to gather operational telemetry data about CPU processing, memory usage, and network transmission while generating detailed logs and metrics and traces, which the system sends through a single data ingestion system. Data preparation for analysis is carried out by the Feature Engineering Layer, which carries out temporal aggregation, normalization, feature extraction, and multimodal fusion. If something out of the ordinary is detected, the Anomaly Detection Layer will use models like Autoencoders and Isolation Forest to find it. Lastly, the Evaluation & Alerting Layer creates notifications for prompt reaction after evaluating performance using accuracy, recall, and F1-score.

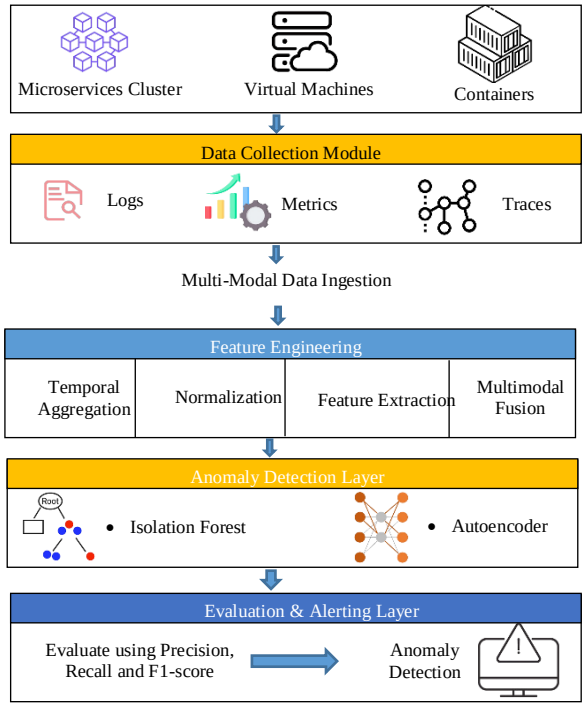

Fig 1: Multi-Modal Anomaly Detection Architecture

3.2. Telemetry Collection

The telemetry collection layer uses a set of cloud-native observability tools to gather system-wide, real-time data. The system receives Kubernetes audit logs via a Kubernetes-native solution that tracks control-plane operations, including API requests and user access patterns. eBPF-based probes are deployed as a DaemonSet in a Kubernetes cluster and collect telemetry data at very low cost, including on a single node. They have proven that record system call events, process

lifecycles and network activities, allowing fine-grained visibility into the runtime behavior of containers.

3.3. Threat Modeling

Threat modeling includes identifying potential attack vectors and defining how malicious behavior may manifest in cloud environments. Common threats include unauthorized access, privilege escalation, resource abuse, and cryptojacking. A baseline of normal system behavior is established by analyzing legitimate user activities, service-to-service communication, and resource utilization patterns under benign conditions. Any deviation from this baseline is treated as a potential anomaly. This process enables the system to capture both known and unknown attack patterns by analyzing variations in user behavior, API calls, and system-level resource usage.

3.4. Feature Engineering and Preprocessing Layer

After telemetry data collection, the data undergoes preprocessing and feature engineering to convert raw, unstructured inputs into structured feature vectors suitable for machine learning models. The process consists of three main steps which involve deleting unnecessary data and combining all information before converting log and metric and trace data into machine-readable formats. The system uses sliding window temporal aggregation to extract time-based patterns while Z-score standardization provides a consistent method to scale different telemetry data sources. The system uses multi-modal feature fusion to combine logs and metrics and traces into a single feature set which allows the model to learn how diverse system parts relate with each other. The system achieves better detection performance and decreases false positive rates within cloud computing environments.

3.5. Anomaly Detection Models

This section presents the machine learning models used for identifying abnormal behavior in cloud telemetry data. The models include Isolation Forest and Autoencoder-based methods.

3.5.1. Isolation Forest

The Isolation Forest algorithm is an unsupervised machine learning method introduced by [9] specifically designed to identify outliers within a dataset. It is predicated on the concept of isolation, in which anomalous observations are characterized by their relative independence from other ob-servations within the feature space. The Isolation Forest algorithm is an unsupervised machine learning method introduced by [9] specifically designed to identify outliers within a dataset. It is predicated on the concept of isolation, in which anomalous observations are characterized by their relative independence from other ob-servations within the feature space.

Anomaly detection is one use case for Isolation Forest, an unsupervised learning system [25]. The idea behind it is "isolation," which means that out-of-the-ordinary observations in the feature space are not related to any other observations. The model calculates the average path length needed to isolate a sample and builds random decision trees. Equation (1) defines the anomalous score:

$$s_{IF}(x) = 2^{-\frac{E(h(x))}{c(n)}} \quad (1)$$

Here, E(h(x)) is the average path length of sample x across isolation trees, and c(n) is a normalization factor that is dependent on the size of the dataset.

3.5.2. Autoencoder

A particular kind of feed-forward neural network in which the input and output are identical is called an autoencoder (AE). The dimensions of the input and output layers are identical. An encoder plus a decoder make comprise an AE.

The input vector is transformed into a latent representation with smaller dimensions during the encoding phase, as demonstrated in Equation (2).

$$z = f_{enc}(x) = \sigma(W_e x + b_e) \quad (2)$$

The original input is reconstructed during the decoding phase using the latent representation described in Equation (3):

$$\hat{x} = f_{dec}(z) = \sigma(W_d z + b_d) \quad (3)$$

Equation (4) defines the reconstruction error, which is employed as a signal for abnormalities Equation (4):

$$E(x) = \left|\left|x - \hat{x}\right|\right|^2 \quad (4)$$

The AE is presented to minimize the E(x) using an unsupervised training method.

3.6. Implementation Details

A scalable observability and machine learning pipeline is used to construct the suggested anomaly detection system in a Kubernetes-based cloud environment. Telemetry data, including logs, metrics, and traces, is collected using OpenTelemetry collectors, streamed via Apache Kafka, and monitored using Prometheus. The raw data is normalized in real time and transformed into structured features through temporal aggregation and multimodal fusion. The system uses containerized microservices to deploy Variational Autoencoders and contrastive learning-based models which train on baseline telemetry data to establish normal behavior patterns. The system analyzes real-time data during inference to generate anomaly scores which it transmits to SIEM systems whenever the scores exceed established thresholds. The system achieves horizontal scalability through Kubernetes HPA while its periodic retraining capabilities enable it to detect concept drift and preserve detection performance.

3.7. Evaluation Metrics

The system's performance was evaluated using standard classification metrics such as recall, precision, F1-score, and ROC curve. In assessment metrics, "True Positive" (TP) refers to positive examples that were correctly anticipated, whereas "False Positive" (FP) refers to negative instances that were wrongly categorized as positive. If a negative event was correctly anticipated, it is called a True Negative (TN); if a positive instance was mistakenly classified as negative, it is called a False Negative (FN). A higher True Positive Rate

(TPR) indicates that more positive occurrences were accurately detected, whereas a lower False Positive Rate (FPR) indicates that more negative examples were mistakenly categorized as positive. The system efficiency assessment uses these essential metrics as its evaluation criteria. Precision represents the proportion of projected anomalies that are genuine positives, whereas recall indicates the proportion of real anomalies predicted by the system. The F1-score provides a reliable evaluation of the system's detection performance; it is a harmonic mean of recall and accuracy. The ROC (Receiver Operating Characteristic) curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across different threshold settings, while the Area Under the Curve (AUC) is a metric that evaluates the model's ability to distinguish between classes. The mathematical representation exists in Equations (5) to (8) which shows the complete mathematical expression:

$$Precision(Prec) = \frac{\text{TP}}{\text{TP+FP}} \quad (5)$$

$$Recall(Rec) = \frac{TP}{TP+FN} \quad (6)$$

$$F1\ score(F1) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

$$AUC = \int_0^1 TPR(FPR)d(FPR) \quad (8)$$

4. Results and Discussion

The proposed anomaly detection system was evaluated in a simulated Kubernetes environment to assess its capability in identifying different types of anomalies, including unauthorized API usage, resource abuse, and container escape attempts. The evaluation was performed using both benign workloads and simulated attack scenarios. The performance of the anomaly detection system across different threat scenarios is summarized in Table II.

Table 2: Performance of the Anomaly Detection System for Different Threat Scenarios

Threat Scenario	Precision	Recall	F1-Score
Resource Abuse	0.91	0.93	0.92
Unauthorized API Use	0.89	0.90	0.89
Container Escape	0.92	0.95	0.94

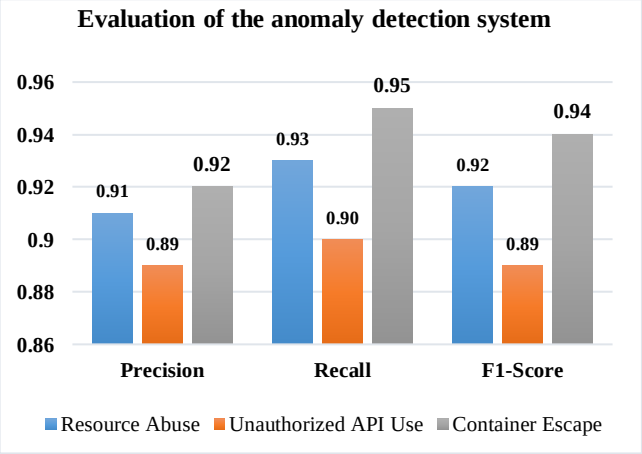

Fig 2: Performance of the Anomaly Detection System for Different Threat Scenarios

The system demonstrated strong performance across both known and unknown attack patterns as visualized in Fig. 2 and Table II. The model achieved its best performance in resource abuse scenarios which included cryptojacking with a prec score of 0.91 and a rec score of 0.93 and an F1 of 0.92. The system detected container escape with an acc of 0.92 and a success rate of 0.95 and an F1 of 0.94. Unauthorized API usage showed slightly lower results which produced a prec score of 0.89 and a rec score of 0.90 and an F1 of 0.89.

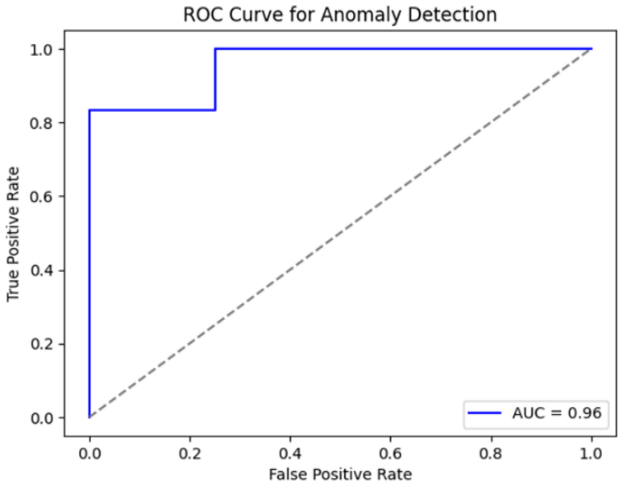

Fig 3: ROC Curve for Anomaly Detection

The ROC curve for the anomaly detection system shows the relationship between TPR and FPR according to the results presented in Fig. 3. The AUC metric showed results above 0.95 for all tested threat scenarios, which demonstrated the system's ability to distinguish normal and abnormal behavior. The model shows strong performance because its high AUC value proves its ability to detect anomalies through different decision threshold levels.

Table 3: Performance Comparison of Anomaly Detection Models across Baseline and Proposed Methods

Model/Threat Scenario	Precision	Recall	F1score
AE[26]	72.63	52.63	61
PCA[27]	75	70	72.41
MLP[28]	77	64	31
Resource Abuse(proposed)	0.91	0.93	0.92
Unauthorized API Use(proposed)	0.89	0.90	0.89
Container Escape(proposed)	0.92	0.95	0.94

Table III presents a comparative evaluation of different anomaly detection models across multiple threat scenarios in cloud environments. The study evaluates resource abuse detection through unauthorized API usage and container escape detection methods which include classical and baseline techniques together with the proposed method. The researchers use precision and recall together with F1-score to measure performance. The results show that traditional models exhibit relatively lower and inconsistent performance while the proposed method demonstrates detection accuracy which exceeds previous results and improved recall and better overall

F1-scores which prove its effectiveness for real-time cloud security anomaly detection tasks.

4.1. Discussion

The proposed anomaly detection system results show its ability to detect various security threats which occur in cloud-native environments. The system achieved high precision and recall results throughout its testing of essential attack situations which included resource abuse through cryptojacking and unauthorized API usage and container escape attempts. The model demonstrated its capacity to identify unusual API behavior through its successful detection of abnormal API activities which resulted in an F1-score of 0.89. The model proved effective through its ability to detect complex and subtle attack patterns.

The proposed approach achieves its main advantage through multi-signal fusion, which combines logs traces and system metrics into a single analytical system. The system achieves better detection capabilities because it can track interdependencies and correlations which usually go unnoticed when individual telemetry sources are examined separately. The model achieves two benefits through its design because it decreases false positives while enhancing its ability to detect anomalies in distributed cloud environments. The study supports recent findings which state that multimodal fusion enhances anomaly detection accuracy for cloud-native systems.

The system includes real-time adaptation features which enable it to manage changing workloads together with concept drift, thus achieving greater system dependability than traditional static detection systems. The system exhibits adaptive capabilities which differ from traditional methods because those methods experience performance decline when system behavior persists through time. The model's AUC values frequently surpass 0.95, which corresponds to earlier hybrid detection study findings. The dynamic nature of cloud environments presents difficulties because organizations need to maintain their ability to track long-term concept drift throughout extended periods. The system needs periodic retraining because its ability to detect threats needs updates whenever new workloads emerge.

5. Conclusion and Future Work

The proposed AI-based system detects anomalies in cloud logs and distributed traces to improve security and reliability of modern cloud-native environments. The system uses a multimodal fusion method to combine logs and metrics and trace data which supports real-time anomaly detection through deep learning and machine learning approaches including Isolation Forest and Autoencoders. Experimental results show strong performance, achieving precision up to 0.92, rec up to 0.95, and an F1 of 0.94 for container escape detection, while resource abuse detection attains prec of 0.91 and rec of 0.93. Unauthorized API usage scenarios achieve an F1-score of 0.89, with overall AUC values consistently above 0.95 across different threat settings. The system uses Kubernetes for its scalable architecture which processes telemetry data in real time to handle various operational demands and multiple system locations. The system achieves strong performance results yet it faces three main challenges which include handling long-term concept drift and enhancing model interpretability and decreasing computational demands during large-scale implementations. The research will develop lightweight AI models which use explainable AI techniques to maintain accurate detection results while decreasing system resource demands and processing time. The research will study reinforcement learning together with self-adaptive systems to support ongoing learning processes in growing cloud computing environments. The research will develop advanced multimodal fusion techniques which combine semantic log enrichment with graph-based trace analysis to achieve better detection performance. The framework can be extended to edge-cloud hybrid environments for broader applicability in next-generation cloud security systems.

References

[1] A. Gupta, “What Is The Right Security Posture? A Perspective on Cloud Computing Security Threats and Risk Assessment,” *Int. J. Emerg. Res. Eng. Technol.*, vol. 4, no. 4, 2023, doi: 10.63282/3050-922X.IJERET-V4I4P112.

[2] S. Bhat, S. R. Sirikonda, V. Katoch, and R. Jain, “Carbon-Kube: A Kubernetes-Native Framework for Multi-Objective Carbon-Aware Scheduling of Big Data Pipelines,” in *2026 9th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, IEEE, Feb. 2026, pp. 1–6. doi: 10.1109/IEMENTech202669403.2026.11434192.

[3] N. Radhasharan, “Real-Time Edge-To-Cloud Intelligence Architecture For Autonomous Drilling Systems,” *J. Int. Cris. RISK Commun. Res.*, vol. 9, no. 1, 2026.

[4] G. Sarraf and V. Pal, “Autonomous Threat Detection and Response in Cloud Security: A Comprehensive Survey of AI-Driven Strategies,” *Int. J. Emerg. Res. Eng. Technol.*, vol. 6, no. 4, 2025, doi: 10.63282/3050-922X.IJERET-V6I4P114.

[5] A. Parupalli and H. Kali, “An In-Depth Review of Cost Optimization Tactics in Multi-Cloud Frameworks,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 5, pp. 1043–1052, Jun. 2023, doi: 10.48175/IJARSCT-11937Q.

[6] S. Singamsetty, “An Intelligent Framework for Secure and Fair Cloud Resource Distribution,” in *2025 7th International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, IEEE, Oct. 2025, pp. 686–690. doi: 10.1109/ICIDCA66325.2025.11280502.

[7] S. K. Chintagunta, “Enhancing Cloud Database Security Through Intelligent Threat Detection and Risk Mitigation,” *TIJER – Int. Res. J.*, vol. 9, no. 10, pp. 49–55, 2022.

[8] C. Carrión, “Kubernetes as a standard container orchestrator-a bibliometric analysis,” *J. Grid Comput.*, vol. 20, no. 4, p. 42, 2022.

[9] B. Singh, H. Singh, and T. Banerjee, “Strengthening Modern IAM Authentication with Quantum Cryptography and Anti-Phishing Techniques,” *Sarcouncil J. Eng. Comput. Sci.*, vol. 4, no. 10, Oct. 2025,

doi: 10.5281/zenodo.17260292.

[10] M. R. Konatham, D. P. Guda, K. Kaushik, W. Sarma, R. Sharma, and M. Soni, "Explainable Deep Learning Framework for Real-Time Threat Hunting and Anomaly Attribution in Enterprise Networks," in *2025 2nd International Conference on Integration of Computational Intelligent System (ICICIS)*, IEEE, Sep. 2025, pp. 1–6. doi: 10.1109/ICICIS65613.2025.11371132.

[11] V. Sharma, "Cloud-Native 5G Deployments: Kubernetes and Microservices in Telco Networks," *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 10, no. 3, pp. 1–8, May 2022, doi: 10.37082/IJIRMPS.v10.i3.232706.

[12] J. Sajja and N. Kolli, "Towards a Unified Framework for Enterprise Data Transformation: Cloud Architecture, Governance, and Intelligent Automation," *J. Inf. Syst. Eng. Manag.*, vol. 9, no. 4, p. 20, 2024.

[13] A. Warrier, "Securing and Scaling API Gateways in Hybrid Environments," *J. Artif. Intell. Mach. Learn. Data Sci.*, vol. 3, no. 3, pp. 2914–2920, Sep. 2025, doi: 10.51219/JAIMLD/Arjun-warrier/607.

[14] D. Bhattacharjee, "Design and Evaluation of Deep Generative AI Model for Intrusion Detection in Cyber Threat Monitoring," in *2025 7th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, IEEE, Dec. 2025, pp. 1–6. doi: 10.1109/ISAECT68904.2025.11318752.

[15] H. P. Cyril, "DeepNetDetect: A Deep Learning-Based Approach for Early Anomaly Detection in Network Traffic," in *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*, IEEE, Feb. 2026, pp. 1–6. doi: 10.1109/ICAIC67076.2026.11395734.

[16] R. Dattangire, R. Vaidya, D. Biradar, and A. Joon, "Exploring the Tangible Impact of Artificial Intelligence and Machine Learning: Bridging the Gap between Hype and Reality," in *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ACET61898.2024.10730334.

[17] K. K. Mohammed, "The Future is Cloud: Modernizing Big Data for the Cloud Era," *Int. J. Sci. Res. Eng. Trends*, vol. 11, no. 5, pp. 1–5, 2025.

[18] J. E. Kofi, "Data-Driven Cloud Workload Optimization Using Machine Learning Modeling for Proactive Resource Management," *Int. J. Emerg. Res. Eng. Technol.*, vol. 6, no. 4, pp. 27–37, 2025, doi: 10.63282/3050-922X.IJERET-V6I4P104.

[19] M. Parikh, A. A. Soni, S. M. Shah, and A. R. Jha, "Big Data Workload Profiling for Energy-Aware Cloud Resource Management." 2026. doi: 10.48550/arXiv.2601.11935.

[20] E. Malul, Y. Meidan, D. Mimran, Y. Elovici, and A. Shabtai, "GenKubeSec: LLM-Based Kubernetes Misconfiguration Detection, Localization, Reasoning, and Remediation," May 2024, doi: 10.48550/arXiv.2405.19954.

[21] M. S. Islam, M. S. Rakha, W. Pourmajidi, J. Sivaloganathan, J. Steinbacher, and A. Miranskyy, "Anomaly Detection in Large-Scale Cloud Systems: An Industry Case and Dataset," Jan. 2025, doi: 10.1109/ICSE-SEIP66354.2025.00039.

[22] S. Potluri, "A Deep Learning-Driven Framework for Detecting Anomalous Data Breaches in Distributed Cloud Storage Infrastructures," *Int. J. Artif. Intell. Data Sci. Mach. Learn.*, vol. 5, no. 3, pp. 80–87, Oct. 2024, doi: 10.63282/3050-9262.IJAIDSML-V5I3P109.

[23] X. Wei *et al.*, "Log-based anomaly detection for distributed systems: State of the art, industry experience, and open issues," *J. Softw. Evol. Process*, vol. 36, no. 8, Aug. 2024, doi: 10.1002/smr.2650.

[24] C. Zhang *et al.*, "DeepTraLog: trace-log combined microservice anomaly detection through graph-based deep learning," in *Proceedings of the 44th International Conference on Software Engineering*, New York, NY, USA: ACM, May 2022, pp. 623–634. doi: 10.1145/3510003.3510180.

[25] M. Almansoori and M. Telek, "Anomaly Detection using combination of Autoencoder and Isolation Forest," in *1st Workshop on Intelligent Infocommunication Networks, Systems and Services*, Online: Budapest University of Technology and Economics, Feb. 2023, pp. 25–30. doi: 10.3311/WINS2023-005.

[26] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, pp. 703–716. doi: 10.1007/978-3-030-30490-4_56.

[27] G. Mattera, R. Mattera, S. Vespoli, and E. Salatiello, "Anomaly detection in manufacturing systems with temporal networks and unsupervised machine learning," *Comput. Ind. Eng.*, vol. 203, p. 111023, May 2025, doi: 10.1016/j.cie.2025.111023.

[28] J. Nobre, E. J. S. Pires, and A. Reis, "Anomaly Detection in Microservice-Based Systems," *Appl. Sci.*, vol. 13, no. 13, 2023, doi: 10.3390/app13137891.