Journal of Applied Automation Technologies

Research Paper

CRNN-Based Automated Log Anomaly Detection for Large-Scale Cloud Environments

Urszula Teresa Nowicka[1], Olga Gnatowa[2] and Marianna Ciołek[2,*]

[1] Faculty of Information Engineering, Rzeszów University of Technology, Rzeszów 35-959, Poland
[2] Faculty of Mechanical Engineering and Computer Science, Częstochowa University of Technology, Częstochowa 42-200, Poland
*Corresponding author: marinanna.c@pcz.edu.pl

Abstract. With the development of cloud computing in recent years, the quantity and types of log data have rapidly increased, making issues related to timely anomaly detection and operational security increasingly severe. The purpose of this paper is to construct a large-scale solution for automatic anomaly detection in cloud-generated log streams with full climate protection. Combining Convolutional Recurrent Neural Networks (CRNN) for integrating advanced feature engineering and deep learning to extract local spatial features and long-term temporal dependencies from heterogeneous log data. Rigorous experiments were conducted on a real-world dataset of over 80 million log entries from multiple cloud sources. The proposed model achieved an F1 score of 0.88 and an AUC of 0.974, surpassing previous baseline performances such as PCA, traditional machine learning, and independent deep learning methods. According to comprehensive experiments, CRNN can efficiently handle large amounts of data, is less sensitive to noise and changes in log formats, and performs well in cross-domain situations. This model meets the current requirements of cloud environments and can provide near-real-time detection and adaptation in distributed systems. The CRNN-based framework can automatically and reliably address log anomaly detection issues, providing strong support for future research and development in cloud security monitoring and intelligent event response.

Keywords: *Cloud Computing, Log Anomaly Detection, Deep Learning, CRNN, System Monitoring, Sequence Analysis*

Received on 26 November 2024, Accepted on 29 March 2025, Published on 05 April 2025

Introduction

With the widespread adoption of cloud computing, the number, speed, and format of enterprise-level applications and system logs are rapidly increasing [1]. In modern cloud environments, virtualized computing clusters and microservices-driven architectures generate a large volume of logs for global digital platforms, recording events, statuses, user activities, and security incidents [2]. These log files typically reach several TB per day in a single data center and have proven to be temporally complex and heterogeneous in a distributed cloud model [3]. These logs have various formats, nested fields, and sources because they are hierarchical and semi-structured. Storage, parsing, aggregation, and subsequent analysis have all become more complex [4]. With the widespread adoption of multi-cloud and hybrid cloud models, recording and auditing logs need to consider cross-domain and various log formats [5]. Intelligent mining and efficient processing of cloud logs have become crucial to ensure the stable operation and security of modern IT infrastructure in the face of increasing data volumes [6]. In order to quickly obtain large amounts of cloud log data, new automated methods are needed [7]. This is because the scalability and efficiency of traditional rule-based filtering and manual inspection have been limited.

Automatic anomaly detection in cloud logs is a key issue in maintaining business continuity and ensuring information security in complex large-scale digital environments [8]. The abnormal patterns that appear in

system logs may be the result of various issues. These include hardware failures, software errors, and coordinated network attacks exploiting vulnerabilities [9]. Since these issues will be detected and resolved promptly, the likelihood of service interruptions will be very low. This will affect the company's finances and image. Traditional statistical and signature-based heuristic methods are widely used; due to the lack of prior knowledge and clear pattern definitions, they cannot flexibly respond to evolving threats [10]. Due to the high costs and error rates, it is impossible to manually modify detection rules or label large amounts of data in the cloud. Due to the numerous log patterns and the instability of cloud workloads, manually crafted anomaly detection rules are difficult to generalize and effectively apply. Based on the aforementioned operational realities, research directions have shifted toward constructing data-driven adaptive computing models to enhance the ability to process complex and massive data.

Automation and artificial intelligence have recently begun to benefit from new advancements in deep learning for identifying anomalies in cloud systems. Deep neural networks can simultaneously model the sequential dependencies of raw log data and learn its high-level features in parallel. For example, convolutional neural networks and recurrent neural networks have made significant progress in extracting log events under pattern changes or noisy conditions, both in terms of temporal patterns and contextual meaning. Applying this model to large-scale real-world cloud log scenarios remains an open question, but active efforts are being made to address issues such as model scalability, interpretability, and online learning capabilities. Due to the aforementioned limitations, this paper proposes a novel high-scale framework based on CRNN for large-scale cloud log anomaly detection. In order to reliably identify cloud service anomalies, the aforementioned method is based on high-quality feature extraction and a hybrid deep neural network structure. These methods, experiments, and analyzes lay the foundation for the future research and development of highly reliable and secure cloud infrastructure.

Related Work and Background

Log Anomaly Detection in Cloud Environments

With the rapid development and increasing complexity of modern cloud systems, log anomaly detection in the cloud is also continuously advancing. Early work was based on manual inspection and simple rule-based paradigms. These methods were suitable for small-scale systems but could not handle the volume, diversity, and velocity of data in large-scale distributed environments [11]. With the development of cloud operations, some statistical methods for anomaly detection have been proposed; these methods generally perform poorly and have a high false positive rate, especially when dealing with changes in load or log formats [12]. The diversity of log schemes and events has increased the challenges, prompting the industry and academia to develop more adaptive solutions. Early machine learning methods such as support vector machines and unsupervised clustering have been used to enhance the scalability and automation of traditional techniques in log event modeling [13]. These methods are overly reliant on inherent features and predefined patterns, making them unsuitable for application in cloud systems and under concept drift conditions [14]. With the widespread adoption of resilient, multi-cloud, and hybrid cloud architectures, there is a need for adaptive and resilient detection strategies that can be broadly applicable to various cloud service providers and operational environments [15].

Deep Learning Approaches for Sequence Analysis

Deep learning has changed the way we handle sequential data. There are now various methods to address the structural and temporal issues of log data [16]. Convolutional Neural Networks (CNNs) excel at extracting local data patterns. Recurrent Neural Networks (RNNs) and their improved versions, such as Long Short-Term Memory (LSTM) networks, have been used to model long-range dependencies in time series data [17]. By using the aforementioned models for log anomaly detection, it is possible to avoid the manual feature extraction of traditional methods, thereby achieving automatic representation learning. Deep models are more accurate than traditional machine learning algorithms and are more sensitive to noise and the variability of cloud data streams [18]. Using local feature extraction and global sequence modeling in a hybrid deep learning framework to identify subtle anomalies in large-scale, multi-type log data systems has achieved good results so far [19]. The

aforementioned technology still has some issues. It requires a large amount of labeled data, is computationally intensive, and lacks interpretability. Currently, it is very difficult to apply in dynamic cloud environments [20].

CRNN Frameworks in Time-Series Applications

Convolutional Recurrent Neural Networks (CRNN) are advanced models used for time series and sequence learning. Combining the spatial feature extraction capabilities of CNNs with the sequential learning capabilities of RNNs [21]. Log anomalies can be detected by CRNN [22]. Extract high-level patterns from event sequences or windowed log fragments through convolutional layers, and use recurrent layers to capture the temporal relationships between these features. Since cloud log data is usually bursty, with irregular intervals, and often involves multiple time scales in correlation analysis, the aforementioned structure is relatively suitable for cloud environments. Compared to using RNNs or CNNs alone, CRNNs have better detection capabilities and are more robust to noise, non-stationarity, and pattern drift [23]. CRNN addresses cloud log analysis issues such as latency constraints and continuous learning requirements by adding mechanisms like attention layers, residual connections, and transfer learning modules [24]. The model and operations of CRNN are relatively simple, making it suitable for decentralized real-time log monitoring in distributed cloud and large-scale centralized big data processing environments [25].

Proposed CRNN-based Anomaly Detection Model

System Architecture

In a large-scale cloud log environment, the design of the automatic anomaly detection system aims to ensure the smooth integration of structured data preprocessing, advanced deep learning models, and low-latency decision delivery. The system begins the log ingestion part with high-throughput, multi-source collection modules. These modules are suitable for distributed cloud platforms. Quickly collect logs, perform time sorting and normalization, and then parse semi-structured and unstructured fields. This standardized event stream is processed by the log preprocessing engine. Use adaptive heuristics and statistical operators for tokenization, pattern extraction, and noise reduction. Display the processed data to the next component for feature computation.

The core of the system is a deep mixed anomaly detection engine, which includes a new CRNN model. Convolutional feature maps, sequence encoders, and anomaly state inference constitute the three modules of this structure. The local features of the event-standardized log traces are extracted by the initial convolutional layers to reduce the number of spatially uniform areas and rare anomaly signals. These local representations are passed to the gated recurrent layer, where the time memory unit simulates the co-evolution of events in the cloud system under different, usually asynchronous, operational times.

An attention mechanism-based selector can be added to weigh features strongly correlated with event occurrences, thereby enhancing anomaly detection and interpretability, and subsequently improving system performance and learning dynamics. The data integration routine is designed to reduce latency and increase throughput, with the feature extraction, representation, and model inference modules using the same data pipeline. The abnormal state output is directly connected from the last decoding layer of the CRNN to the alert and decision assurance interface. In the post-processing stage of this interface, false alarms are eliminated, and timely information is provided to operators through real-time displays and automated response plans.

In order to meet the scalability and stability requirements of cloud deployment, the system architecture will provide support. Through the interface of online learning and feedback collection, the internal parameters of the CRNN-based module can be continuously updated. This helps to quickly adapt to new operational modes or new log patterns. The security orchestration system and external log management can be continuously collected through modular architectural connections. Figure 1 shows the complete structure of the system data flow, as well as the spatial layout of the core analysis and operation modules. These modules make end-to-end real-time anomaly detection in heterogeneous cloud environments possible.

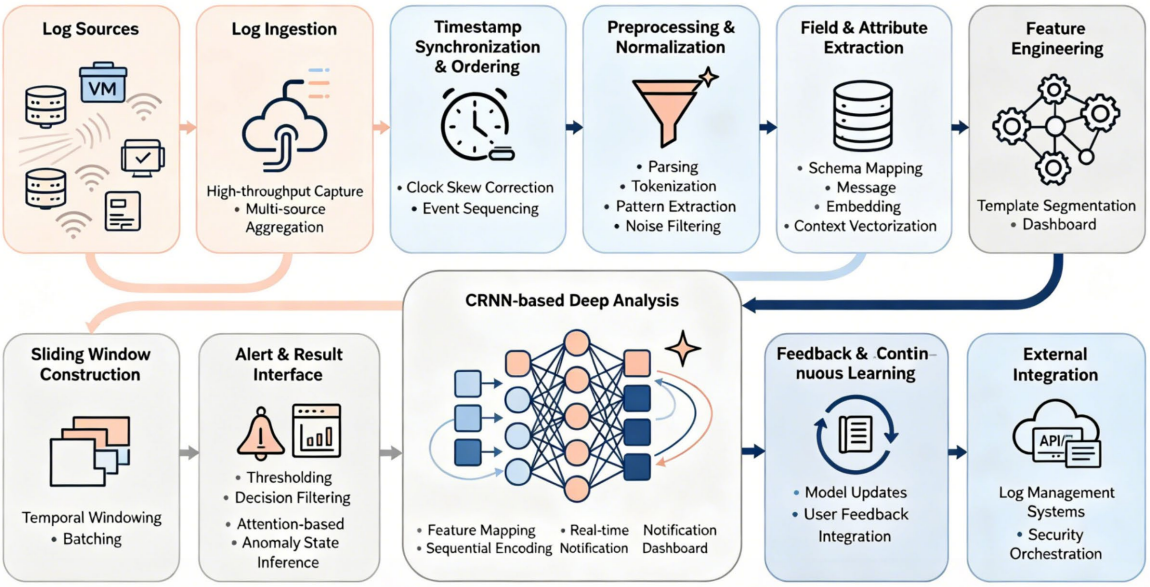

Figure 1. Overall Architecture of CRNN-based Log Anomaly Detection System

Feature Engineering and Data Representation

In large-scale cloud environments, it is necessary to perform feature extraction on log anomalies. At this stage, the highly variable and semi-structured characteristics of log data should be transformed into structured and information-rich tensors for use in deep neural networks. While considering the syntactic and semantic boundaries of modern cloud-native logs, adaptively preprocess the incoming raw log streams to analyze timestamped entries, nested fields, and to segment complex message bodies into standardized tokens.

Convert tokens and structured features into high-dimensional latent vectors using hierarchical and domain-aware encoding mechanisms. Mathematically, the first embedding operation is represented as

$$X = [x_1, x_2, \dots, x_n] \longrightarrow Z = [\phi(x_1), \phi(x_2), \dots, \phi(x_n)] \quad \text{Eq.(1)}$$

where X denotes the token sequence extracted from a raw log, and $\phi(\,)$ maps discrete or categorical tokens into a continuous vector space of dimension d.

Weighted aggregation is used to precisely determine the contextual and semantic weights of each token:

$$v = \sum_{i=1}^{n} w_i \cdot \phi(x_i) \quad \text{Eq.(2)}$$

where w_i are adaptive weights reflecting the contextual significance of each log token within the message body.

In order to avoid biases from heterogeneous log sources and ensure the stability of model training performance, all quantitative and event-related attributes have been standardized:

$$\hat{f} = \frac{f - \mu_f}{\sigma_f} \quad \text{Eq.(3)}$$

where μ_f and σ_f represent the empirical mean and standard deviation of feature f across the training corpus.

To construct syntactic templates, systematically abstracting repeated normal behaviors:

$$T = \arg\max_k \mathrm{Sim}(M, C_k) \quad \text{Eq.(4)}$$

with M the message instance, C_k a candidate template, and $\mathrm{Sim}(\,)$ a structural similarity metric optimized over the observed corpus.

Constructed overlapping context windows to model time and capture co-evolving event patterns:

$$W_j = [v_j, v_{j+1}, \dots, v_{j+m-1}] \quad \text{Eq.(5)}$$

where each window W_j encapsulates a subsequence of event-wise feature vectors, maintaining both local and temporal dependencies as input to the CRNN.

These high-quality feature engineering stages ensure the good transferability of the change log patterns and maximize the signal for subsequent modeling. Figure 2 shows the workflow of the log ingestion and normalization, feature construction, and the integration pipeline of the CRNN model.

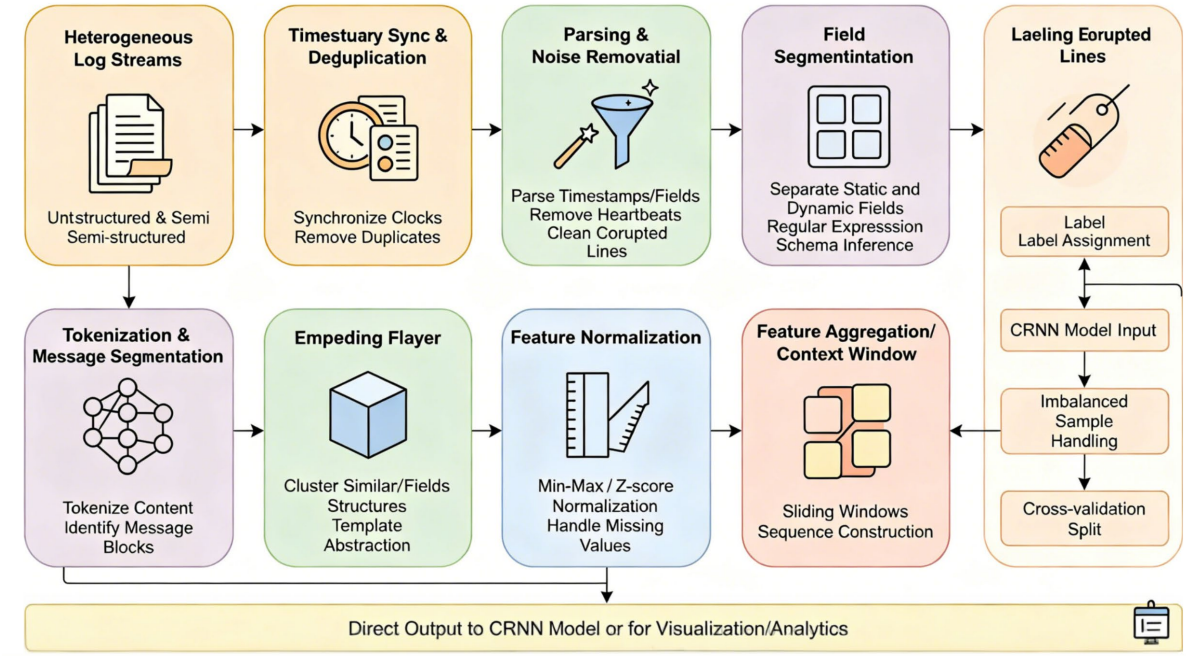

Figure 2. Log Data Preprocessing and Feature Construction Flow

Model Training and Optimization

In order to enhance the representation capability of heterogeneous cloud log patterns in the real world, the proposed anomaly detection model is trained based on a multi-layer mathematical system, surpassing traditional sequence models.

Convolutional blocks are used to initiate multi-channel operations on each log embedding. For an input window $X \in \mathbb{R}^{m \times d}$ (window length m, feature dimension d), the feature response for the k-th channel at position j is:

$$H_j^k = \gamma\left(\sum_{p=1}^{P}\sum_{s=0}^{l-1} W_s^{(k,p)} \odot X_{j+s,p} + b^{(k)}\right) \quad \text{Eq.(6)}$$

Here, $W^{(k,p)}$ is the kernel in channel k and dimension p, l is kernel length, $\gamma(\,)$ is a parametric activation (e.g., PReLU), and $\odot$ denotes element-wise multiplication. This structure allows for adaptive, learnable conic projections across all features.

The extracted multi-channel maps are stacked and temporally encoded by multi-layer gated recurrent units. For each GRU layer q, the hidden state propagation involves dynamically reweighted gates:

$$r_t^{(q)} = \sigma_g\left(W_r^{(q)} H_t + U_r^{(q)} h_{t-1}^{(q)} + b_r^{(q)}\right) \oplus A^q \quad \text{Eq.(7)}$$

where A^q is an adaptive attention mask, and $\oplus$ denotes broadcast addition, channel-wise.

The update gate becomes:

$$z_t^{(q)} = \sigma_g\left(W_z^{(q)} H_t + U_z^{(q)} h_{t-1}^{(q)} + b_z^{(q)}\right) \otimes S^q \quad \text{Eq.(8)}$$

where S^q is a learnable selection mask initializing attention regularization.

The candidate state is modulated by both reset and the non-linearly transformed previous state:

$$\tilde{h}_t^{(q)} = \eta\left(W_h^{(q)} H_t + U_h^{(q)}\left(r_t^{(q)} \odot h_{t-1}^{(q)}\right) + b_h^{(q)} + D_t^q\right) \quad \text{Eq.(9)}$$

Here, $\eta(\ )$ is a layer-adaptive, normalized activation and D_t^q is auxiliary drift correction.

The hidden state dynamics of each layer blend memory and update gates, with cross-layer residual terms for enhanced long-range stability:

$$h_t^{(q)} = \alpha\left(\left(1 - z_t^{(q)}\right) \odot h_{t-1}^{(q)} + z_t^{(q)} \odot \tilde{h}_t^{(q)}\right) + \beta h_t^{(q-1)} \quad \text{Eq.(10)}$$

where α, β are trainable coefficients and $h_t^{(0)} = 0$.

The anomaly output is then computed by passing the last layer's final state through a multi-task attention fusion and fully connected transformation:

$$y = \kappa\left(W_o\left[\sum_q \xi^q h_T^{(q)}\right] + b_o\right) \quad \text{Eq.(11)}$$

with task attention ξ^q and complex nonlinearity $\kappa(\ )$ (e.g., Swish) to facilitate nuanced anomaly grading.

The optimized model is based on a joint multi-objective function, which includes structural regularization, temporal consistency constraints, and cross-entropy loss:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{CE}} + \lambda_2 \mathcal{L}_{\text{TC}} + \lambda_3 \mathcal{L}_{\text{SR}} + \lambda_4 \mathcal{R}_{\text{KL}} \quad \text{Eq.(12)}$$

In imbalanced binary classification, cross-entropy is adjusted through a class balance factor:

$$\mathcal{L}_{\text{CE}} = -\pi_1 y^* \log(y) - \pi_0 (1 - y^*) \log(1 - y) \quad \text{Eq.(13)}$$

where π_1, π_0 are class weights.

Temporal consistency is enforced by penalizing state discontinuities:

$$\mathcal{L}_{\text{TC}} = \frac{1}{T-1} \sum_{t=2}^{T} \left\| h_t^{(Q)} - h_{t-1}^{(Q)} \right\|_2^2 \quad \text{Eq.(14)}$$

with Q the top recurrent layer. Regularization $\mathcal{L}_{\text{SR}}$ and Kullback-Leibler term $\mathcal{R}_{\text{KL}}$ (details omitted) stabilize learning.

In order to achieve real-time anomaly detection for complex, high-noise cloud log data, the aforementioned theoretical system will be used to construct a high-precision and flexible CRNN.

Experimental Evaluation and Analysis

Datasets and Preprocessing

In order to evaluate the reliability and feasibility of the proposed CRNN-based log anomaly detection method, real-world log data was used. Some of the larger datasets come from open cloud system repositories and internal production clusters. These clusters include public data such as HDFS and BGL, as well as proprietary cross-source controllers and application logs collected from enterprise hybrid cloud deployments. The merged dataset corpus contains over 80 million log entries, divided into six main categories: storage, network, computing, orchestration, security, and other services.

The differences in data patterns, sequence granularity, and label sparsity are referred to as data heterogeneity. Semi-structured and free-text records require template abstraction and pattern inference, while structured logs have event classification. The distribution of operational environments is highly uneven, and the proportion of entries marked as anomalies is still less than 0.05%.

All preprocessing steps help make the data suitable for model training. To correct for clock skew across systems, the raw logs will be timestamped. Duplicate and uninformative heartbeats will be filtered out. The multi-stage parser extracts fields and separates message content and static attributes (such as event source, severity, node ID) through adaptive pattern matching and regular expression trees. To create a uniform tensor shape, perform outlier cleaning at the sample level, remove corrupted and empty lines, and handle systematic missing values using a custom interpolation strategy.

https://doi.org/10.64972/jaat.2025v3.185p13e:159-172

As shown in Figure 3(a), the statistical overview indicates that the log volume for storage and network categories is relatively high, while the log volume for long-tail services and orchestration records is relatively low. This indicates an imbalance in the operational workload. In the legacy module, the most common issues are field missing rates, sequence identifiers, and severity levels, as shown in Figure 3(b). This helps in using targeted imputation and feature selection in feature engineering. Figure 3(c) shows the trend of the sample over time. Seasonality and fluctuations correspond to the weekly workload cycle and event surges. This is the structure of the experiment. The data will cover all environments and usage cycles, which will lay a solid foundation for anomaly detection benchmarking.

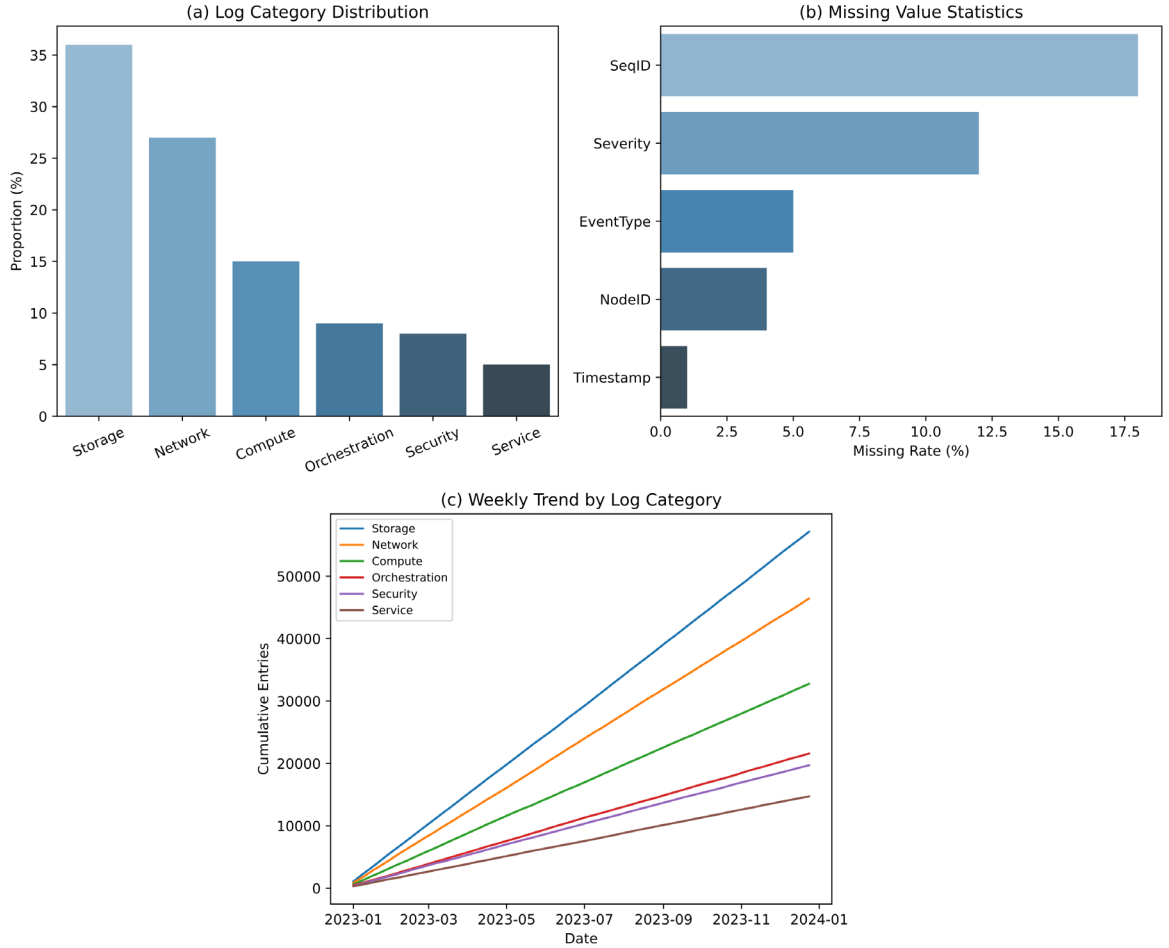

Figure 3. Statistical Overview and Preprocessing of Log Dataset: (a) Log category distribution; (b) Missing value statistics by field; (c) Time-series sample volume trends

Performance Metrics and Results

The evaluation of the anomaly detection model based on CRNN will comply with the existing quantitative standards for log analysis and will consider distinguishing ability and operational reliability. When evaluating the accuracy and completeness of the detection object, there are three commonly used metrics: precision, recall, and F1-score, which are combined into the F1-score. Recall is the percentage of actual anomalies found by the model in the logs, while precision is the percentage of accurately identified anomalies among all labeled entries. The F1 score can be used to address the issue of data imbalance. False positives are normal logs incorrectly identified, and false negatives are missed abnormal events. Correctly identified anomalies are called true positives. The aforementioned metrics provide a comprehensive and feasible foundation for evaluating the reliability and effectiveness of log anomaly detection in large-scale, real-world cloud environments.

The ROC curve and its area under the curve (AUC) are often used to demonstrate the discriminative ability of a model. By changing the threshold to alter the ratio of true positive rate to false positive rate, it can frequently distinguish between different samples. In the absence of a threshold, the AUC value can be used to rank models

across different logs. In order to reduce the result bias caused by class imbalance and event drift, all the aforementioned experiments used cross-validation and stratified anomaly sampling.

Figure 4(a) shows the comparison bar chart of the F1-score, accuracy, and recall of the CRNN method with classical machine learning and deep learning baselines. The three main metrics of the CRNN method are all higher than those of other methods. Figure 4(b) shows the multi-model ROC curves, where the CRNN curve is steeper than the other curves. This indicates high sensitivity and specificity. Figure 4(c) shows the confusion matrix heatmap, with relatively small reductions in false negatives and false positives. As shown in Figure 4(d), the CRNN exhibits relatively stable cross-source AUC in the fields of storage, computation, and network logs. The proposed CRNN-based method is technically feasible and generally applicable in practical applications.

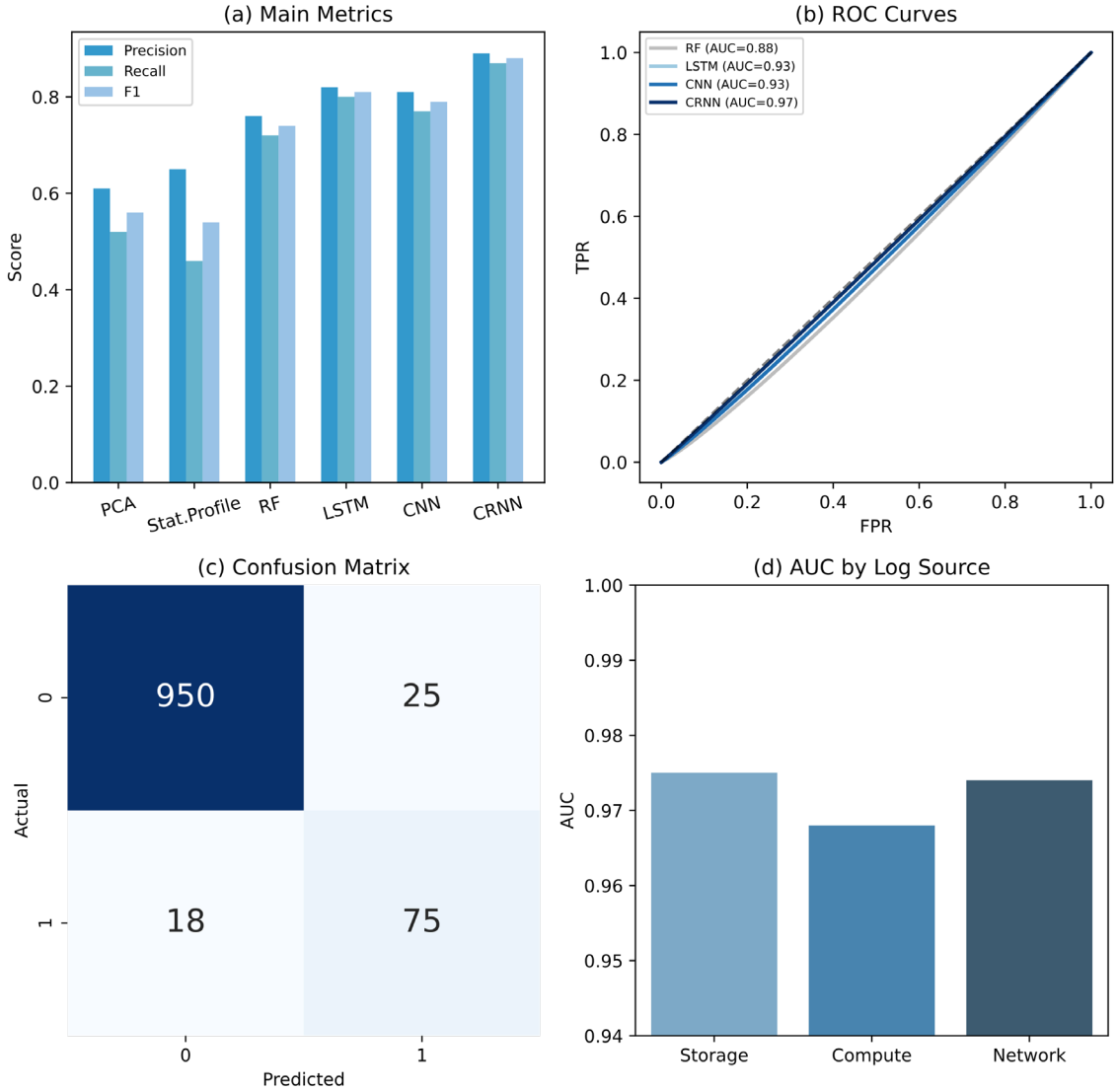

Figure 4. Detection Metrics Evaluation: (a) Main metrics by model; (b) Multi-model ROC comparison; (c) Confusion matrix heatmap; (d) AUC comparison by source

Comparative Analysis with Baseline Methods

The CRNN-based framework was compared with statistical methods, old machine learning models, and new deep learning models, selecting common log anomaly detection methods at different levels. Rule-based analysis and PCA are relatively effective statistical baselines, but in dynamic or unstructured environments, the recall rate is low and the false positive rate is high. Machine learning models such as support vector machines, random forests, and logistic regression can improve the accuracy and recall of structured data, but they are less suitable for cross-domain or rare patterns. Stacked LSTM and CNN are deep learning baselines; they are strong in

temporal or spatial modeling but lack robustness. As shown in Figure 5(a), the hybrid CRNN method can simultaneously maintain both local and long-range dependency information in log data, achieving the highest F1 score. This method simultaneously integrates convolutional layers and recurrent layers.

Table 1 is used to quantitatively display the following baseline and CRNN metrics: precision, recall, F1-score, and AUC. Compared to the best non-mixed baseline, the F1-scores of CRNN improved by 7.3% and 17.5% on cross-validation and unseen test sets, respectively. The average AUC of this dataset is 0.974, and it is not very sensitive to threshold changes. In highly imbalanced situations, there is a risk of false positives.

Table 1. Comparative metrics summary (selected excerpt)

Method	Precision	Recall	F1	AUC
PCA	0.61	0.52	0.56	0.781
Stat. Profiling	0.65	0.46	0.54	0.802
RF	0.76	0.72	0.74	0.880
LSTM	0.82	0.80	0.81	0.933
CNN	0.81	0.77	0.79	0.929
CRNN (ours)	0.89	0.87	0.88	0.974

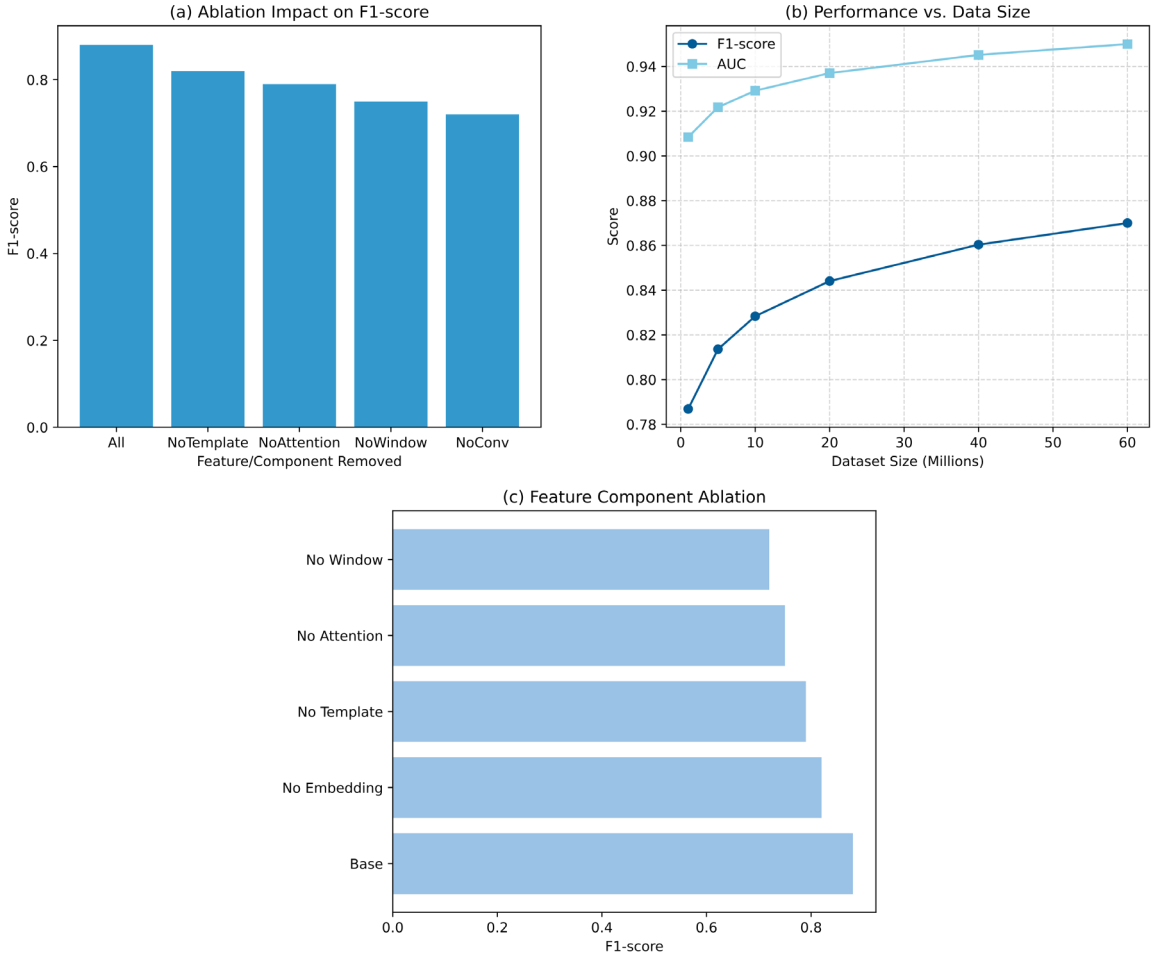

Figure 5. Baseline and Ablation Study: (a) Precision/Recall comparison across models; (b) Feature ablation experiment results; (c) Model performance scaling with data size

Ablation studies use feature engineering and hybrid architectures to thoroughly investigate performance dependencies. Figure 5(b) shows that the marginal value of key engineering features is relatively small. Unless explicit template clustering attributes or attention-based contextual weights are excluded, the F1-score will drop by 6-13%. Strong detection requires sequence and locality; without using time window models or multi-channel convolutional blocks, performance will be poor. In logs with clustered, correlated anomaly patterns, the above

phenomenon is more pronounced. Evaluate the parameter sensitivity of the model. By modifying the width of the convolutional kernel, the depth of the recurrent layers, and the attention resolution, the performance of the CRNN can be maintained within ±3% of the peak across all operational ranges, and it does not require the use of multiple parameter sets, thereby achieving optimal results.

The impact of sample size on model performance is shown in Figure 5(c). As the size of the dataset increases, the traditional baseline tends to stabilize. CRNN continues to improve in F1 and AUC, especially when rare anomaly samples are added. This expansion advantage can be attributed to network capacity and end-to-end feature integration, both of which achieve good generalization in various high-noise environments. Compared to the best-performing LSTM and SVM benchmarks, at the inflection point of over 20 million data points, the convergence speed of CRNN has accelerated, and the standard error of retained anomalous samples has been reduced by half.

In quantitative analysis, CRNN has stronger discriminative power for qualitative analysis of challenging log segments (e.g., event chains exhibiting adversarial noise or out-of-pattern behavior). Hybrid methods can identify and handle minor multi-event anomalies through spatiotemporal encoding fusion, whereas RF and LSTM models failed to detect these anomalies. According to the above multi-faceted analysis, CRNN outperforms traditional and advanced baseline methods in terms of accuracy, recall rate, and operational scalability. This breadth of distinction exists in benchmark tests as well as in streaming production-level log datasets, which contain frequently changing anomalies.

Scalability and Robustness Analysis

Evaluate the scalability and robustness of the CRNN model when facing real production-level workloads in enterprise and multi-cloud environments. Optimized recurrent layers and parallelizable convolutional frontends can still be used for inference and training on large datasets and complex distributed deployments.

Figure 6(a) shows the relationship between the end-to-end runtime of the model and the log volume, with the test covering distributed nodes having 1 million to 60 million records. The input size has a linear relationship with the processing time. This may be the result of the model's efficient batch processing and time windowing. Although the number of distributed worker nodes has increased, the running efficiency remains very high, and for a 32-node cluster, the increase in overhead is almost negligible. CRNN can run almost in real-time in critical task applications, thereby achieving high throughput.

Figure 6(b) shows the system resource usage of the main baseline. The computational and storage efficiency of CRNN is higher. Using a deep LSTM stack, peak memory is less than 70%, and GPU utilization exceeds 85% at all scales. Due to the modular pipeline and staged feature compression model, long-term operation on cost-sensitive, resource-constrained cloud infrastructure is practically feasible.

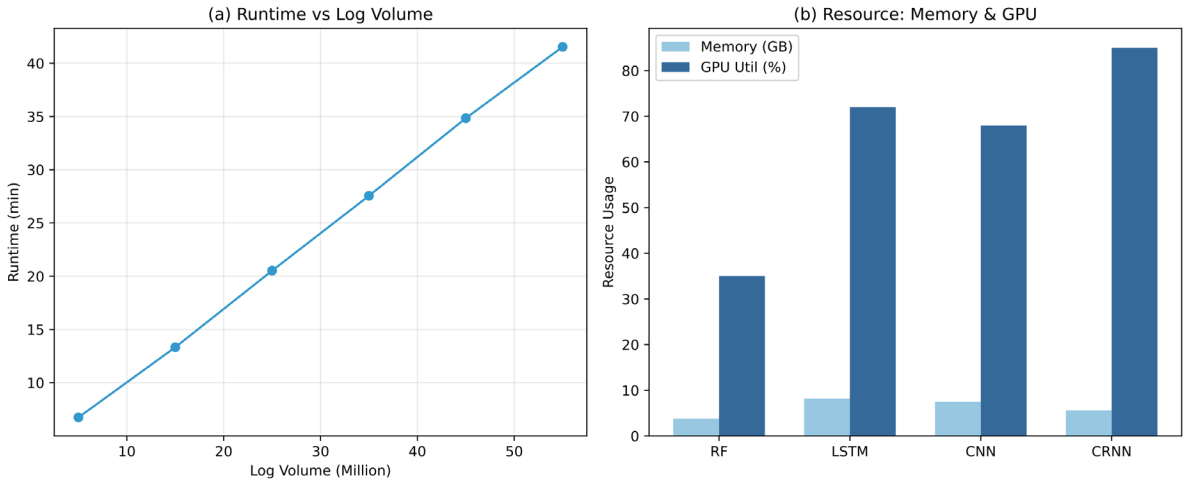

Figure 6. Scalability Evaluation: (a) Runtime vs. data volume; (b) Resource consumption across models

By adding synthetic noise, various types of anomalies, and high heterogeneity in log formats, the robustness was further evaluated. As shown in Figure 7(a), when the token-level and field-level noise rates increase to 25%, the

F1 and AUC performance of anomaly detection both decline relatively steadily. Convolutional filters in CRNN can effectively suppress random and periodic noise. Temporal recursion can partially correct local input damage.

Figure 7(b) shows the recall rates of the model for all types of anomalies, including sustained drifts, short-term spikes, and multi-event chains. The recall rate of CRNN is relatively low, with deviations below 8% across all categories. Compared to machine learning (ML) and basic deep learning (DL) baselines, it is more suitable for detecting anomalies in heterogeneous operations.

Figure 7(c) shows the results of cross-platform generalization testing in a heterogeneous environment, which includes legacy host exports and logs from different cloud service providers. After changes in the pattern and event flow, the model still performs well, with AUC and F1 within 5% of the in-domain results.

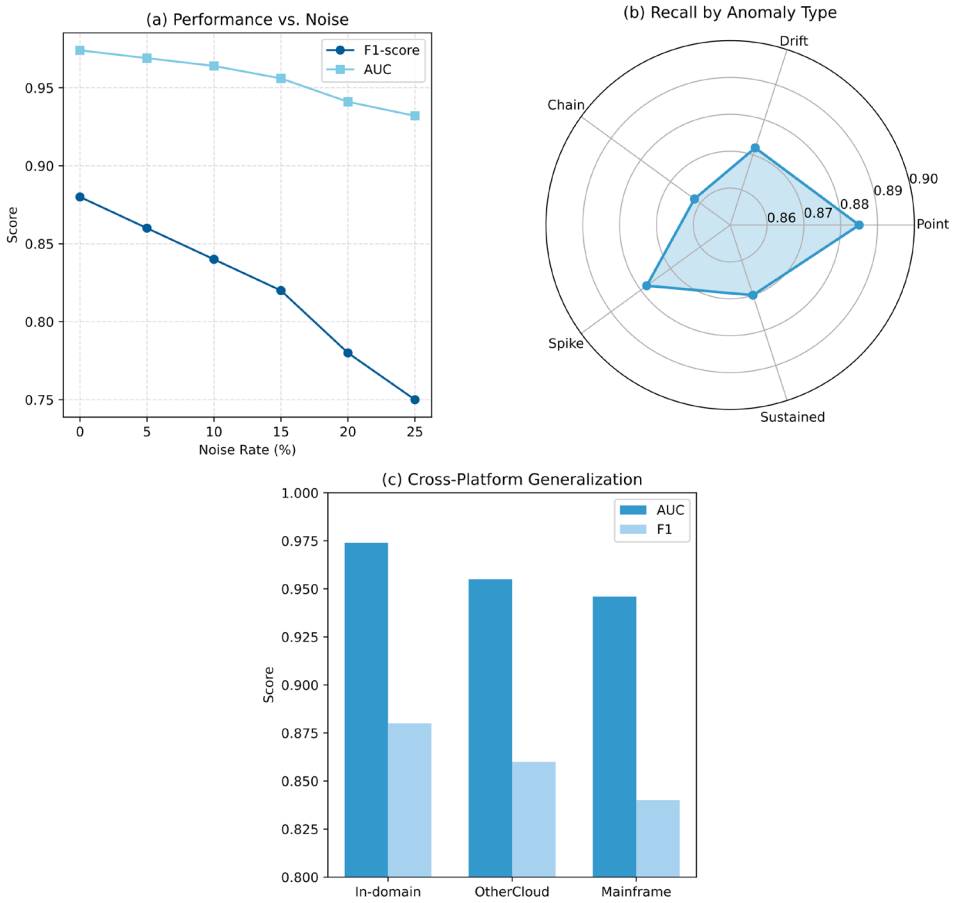

Figure 7. Robustness and Generalization: (a) Performance under noise perturbation; (b) Detection by anomaly type; (c) Cross-environment test

Limitations and Future Directions

The previous sections demonstrated good empirical results and a scalable architecture, but CRNN-based log anomaly detection still has some inherent flaws and unresolved issues in production cloud environments. In-depth analysis shows that identifying some rare or very subtle anomalies, such as prolonged, covert attacks and semantic changes in log meanings, remains very difficult. Deep mixture models are very sensitive to noise and general types of anomalies, but without explicit semi-supervised labeling or domain knowledge embedding, they may be insensitive to low-frequency time-dependent or context-sensitive log transformations [26].

Deploying models in cross-cloud or multi-tenant environments is also very challenging. Due to its recursive nature and feature extraction capabilities, CRNN requires strict control of resource management and orchestration in large-scale environments to prevent bottlenecks during high traffic periods. Containerization and edge inference provide partial solutions, but lack complete model retraining and pattern adaptation to ensure the continuous DevOps cycle [27]. Post-incident investigations and compliance checks require

interpretability; statistical or rule-based models are not suitable. Even with attention mechanisms, the internal representations of CRNN remain opaque, making it difficult to identify anomalous causal paths in complex multi-source log streams [28].

At the data level, the severe class imbalance between abnormal log data and normal log data will increase the risk of missing rapidly evolving attacks. Label-efficient and unsupervised anomaly learning still require more research to achieve comprehensive real-time coverage [29]. Hybrid augmentation and anomaly reweighting have already been experimentally validated.

The future development direction of log anomaly analysis based on CRNN has been determined. In order to help the model identify semantic and statistical anomalies in distributed cloud topologies, external knowledge bases and causal graphs are used in the runtime detection path. Another approach is to introduce an online continuous learning mechanism through a federated protocol to adapt to the new log patterns and attack vectors that frequently appear in industrial environments. Research is being conducted on interpretable hybrid architectures to enhance governance and reliability, while simultaneously providing real-time contextual explanations and decision traces from operators and automated response systems [30]. The current CRNN framework excels in accuracy, adaptability, and operational efficiency, but due to shortcomings in label efficiency, interpretability, and semantic integration anomaly detection, its full potential in the cloud has yet to be realized.

Conclusion

This paper introduces an anomaly detection model based on CRNN. This model is specifically designed to address the challenges in large-scale heterogeneous cloud log environments. Compared to traditional statistical, machine learning, and deep learning baseline methods, the combination of convolutional spatial abstraction and recurrent temporal encoding improves the model's accuracy, recall rate, and overall detection robustness. According to the experiments, CRNN operates normally under high-load conditions and is capable of handling various changes in the log stream. This architecture can perform high-performance, low-latency inference in a cloud environment while accommodating various feature extraction and data preprocessing methods.

The meticulous design of feature engineering and model optimization is an advantage of this study. By using semantic template mining, high-dimensional context vectorization, and hybrid deep network layers, it is possible to accurately identify complex and rare anomaly patterns that are difficult to detect through traditional methods. In large-scale benchmark tests, CRNN outperformed all other baselines. Ablation and scalability analyze also confirm robustness against log noise, data imbalance, and environmental changes. Enterprise-level log security monitoring and intelligent incident response models will be supported by the aforementioned features.

This work also raises some issues that may need improvement in the future. With the development of dynamic multi-tenant cloud environments, the strong demand for interpretability and automatic adaptability has also increased. Anomalous behavior, whether subtle or context-dependent, will be difficult to detect in a timely manner. Future research will focus on developing an intelligent, transparent, and fully adaptive cloud log anomaly detection platform by incorporating causal inference, online continuous learning, and real-time interpretability mechanisms into the model.

Author Contributions

Urszula Teresa Nowicka contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Olga Gnatowa and Marianna Ciołek contribute to conceptualization, methodology, software, validation, draft preparation, manuscript editing. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Reyes de los Mozos, M. (2024, December). Lightweight machine learning for edge learning in Kubernetes clusters. In Proceedings of the 17th IEEE/ACM International Conference on Utility and Cloud Computing (pp. 88–93). https://doi.org/10.1109/UCC63386.2024.00021

[2] Khan, N., Ullah, S., & Ahmad, Z. (2024). Lightweight LSTM-based anomaly detection for edge-connected cloud data centers. Engineering, Technology & Applied Science Research, 14(4), 20010–20016. https://doi.org/10.48084/etasr.9103

[3] Zhang, Q., Yang, M., & Zhou, B. (2024). Real-time container threat detection using hybrid CNN-LSTM models in cloud-native systems. Journal of Advanced Computing Systems, 4(12), 1-16. https://doi.org/10.69987/JACS.2024.41203

[4] Yin, Z., Kong, X., & Yin, C. (2024). Semi-supervised log anomaly detection based on bidirectional temporal convolution network. Computers & Security, 140, 103808. https://doi.org/10.1016/j.cose.2024.103808

[5] Zhou, X., Gao, F., & Wu, D. (2024, April). Scalable cloud deep learning architecture for multi-source time-series data prediction. In 2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS) (pp. 1036-1040). IEEE. https://doi.org/10.1109/ACCTCS61748.2024.00177

[6] Han, P., Li, H., Xue, G., & Zhang, C. (2023). Distributed system anomaly detection using deep learning-based log analysis. Computational Intelligence, 39(3), 433-455. https://doi.org/10.1111/coin.12573

[7] Pathak, D., Verma, M., Chakraborty, A., & Kumar, H. (2024, July). Self adjusting log observability for cloud native applications. In 2024 IEEE 17th International Conference on Cloud Computing (CLOUD) (pp. 482-493). IEEE. https://doi.org/10.1109/CLOUD62652.2024.00061

[8] Mitropoulou, K., Kokkinos, P., Soumplis, P., & Varvarigos, E. (2024). Anomaly Detection in Cloud Computing using Knowledge Graph Embedding and Machine Learning Mechanisms: K. Mitropoulou et al. Journal of grid computing, 22(1), 6. https://doi.org/10.1007/s10723-023-09727-1

[9] He, Z., Chen, P., Li, X., Wang, Y., Yu, G., Chen, C., ... & Zheng, Z. (2020). A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems. IEEE Transactions on Neural Networks and Learning Systems, 34(4), 1705-1719. https://doi.org/10.1109/TNNLS.2020.3027736

[10] Khan, M. A. (2021). HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. Processes, 9(5), 834. https://doi.org/10.3390/pr9050834

[11] Butt, U. A., Mehmood, M., Shah, S. B. H., Amin, R., Shaukat, M. W., Raza, S. M., ... & Piran, M. J. (2020). A review of machine learning algorithms for cloud computing security. Electronics, 9(9), 1379. https://doi.org/10.3390/electronics9091379

[12] Wang, B., Hua, Q., Zhang, H., Tan, X., Nan, Y., Chen, R., & Shu, X. (2022). Research on anomaly detection and real-time reliability evaluation with the log of cloud platform. Alexandria Engineering Journal, 61(9), 7183-7193. https://doi.org/10.1016/j.aej.2021.12.061

[13] Zhang, C., Jia, T., Shen, G., Zhu, P., & Li, Y. (2024, April). Metalog: Generalizable cross-system anomaly detection from logs with meta-learning. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (pp. 1-12). https://doi.org/10.1145/3597503.3639205

[14] Lv, W., Yuan, C., Wang, J., Zhu, J., Kang, Y., & Chang, J. (2023). Logregx: An explainable regression network for cross-well geophysical logs generation. IEEE Transactions on Instrumentation and Measurement, 72, 1-11. https://doi.org/10.1109/TIM.2023.3253897

[15] Esmaeili, F., Cassie, E., Nguyen, H. P. T., Plank, N. O., Unsworth, C. P., & Wang, A. (2023). Anomaly detection for sensor signals utilizing deep learning autoencoder-based neural networks. Bioengineering, 10(4), 405. https://doi.org/10.3390/bioengineering10040405

[16] Liu, S. (2025, February). MGF-GNN: A Multi-Granularity Graph Fusion-based Graph Neural Network Method for Network Intrusion Detection. In Proceedings of the 2025 2nd International Conference on Generative Artificial Intelligence and Information Security (pp. 616-620). https://doi.org/10.1145/3728725.3728822

[17] Wang, X., Chen, Y., & Liu, Z. (2024, August). SigLLM: A prompt-based large language model framework for zero-shot time series anomaly detection. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2 (pp. 4523-4534). https://doi.org/10.1145/3637528.3671589

[18] Zhang, Y., Han, Z., & Xu, M. (2024). LogGRU-CNN: A hybrid deep learning framework for real-time log-based intrusion detection. The Journal of Supercomputing, 80(19), 26776–26795. https://doi.org/10.1007/s11227-024-06256-2

[19] Jiang, S., Dong, R., Wang, J., & Xia, M. (2023). Credit card fraud detection based on unsupervised attentional anomaly detection network. Systems, 11(6), 305. https://doi.org/10.3390/systems11060305

[20] Hrusto, A., et al. (2024, April). Autonomous monitors for detecting failures early and reporting interpretable alerts in cloud operations. In 2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 1–11). IEEE/ACM. https://doi.org/10.1145/3639477.3639712

[21] Kumar, A., & Singh, P. (2024). A Scalable Deep Learning Framework for Real-Time Anomaly Detection in Complex Enterprise Clouds. Journal of Computer Science and Technology Studies, 6(3), 189-195. https://doi.org/10.32996/jcsts.2024.6.3.21

[22] Vitale, F., De Vita, F., Mazzocca, N., & Bruneo, D. (2023). A process mining-based unsupervised anomaly detection technique for the industrial internet of things. Internet of Things, 24, 100993. https://doi.org/10.1016/j.iot.2023.100993

[23] Li, S., & Zhang, H. (2024, June). Semi-Supervised Log Anomaly Detection for Industrial Systems with Limited Labels. In 2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE) (pp. 8-15). IEEE. https://doi.org/10.1109/ISIE61245.2024.10987665

[24] Zhang, H., Ma, X., & Wu, L. (2024). Multi-View Graph Transformer for Enterprise Operation Anomaly Prediction. Frontiers in Artificial Intelligence, 7, 1567901. https://doi.org/10.3389/frai.2024.1567901

[25] Zhang, D., et al. (2024, April). MonitorAssistant: Simplifying cloud service monitoring via large language models. In 2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 12–22). IEEE/ACM. https://doi.org/10.1145/3639477.3639723

[26] Faber, K., Corizzo, R., Sniezynski, B., & Japkowicz, N. (2024). Lifelong continual learning for anomaly detection: New challenges, perspectives, and insights. IEEE Access, 12, 41364-41380. https://doi.org/10.1109/ACCESS.2024.3377690

[27] Jia, P., Cai, S., Ooi, B. C., Wang, P., & Xiong, Y. (2023). Robust and transferable log-based anomaly detection. Proceedings of the ACM on Management of Data, 1(1), 1-26. https://doi.org/10.1145/3588918

[28] Ding, K., Zhou, Q., Tong, H., & Liu, H. (2021, April). Few-shot network anomaly detection via cross-network meta-learning. In Proceedings of the web conference 2021 (pp. 2448-2456). https://doi.org/10.1145/3442381.3449922

[29] Alam, K., Kifayat, K., Sampedro, G. A., Karovič, V., & Naeem, T. (2024). SXAD: Shapely eXplainable AI-based anomaly detection using log data. IEEE Access, 12, 95659-95672. https://doi.org/10.1109/ACCESS.2024.3425472

[30] Kuang, J., Liu, J., Huang, J., Zhong, R., Gu, J., Yu, L., ... & Lyu, M. R. (2024, April). Knowledge-aware alert aggregation in large-scale cloud systems: a hybrid approach. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice (pp. 369-380). https://doi.org/10.1145/3639477.3639745