ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

ML Based Intelligent Log Analytics Using BERT

[1] Nishanth V, [2] Kanchana T, [3] Rajamanickam V, [4] Badrinath C, [5] Sathish V
[2] Assistant Professor (CSE) Department of Computer Science and Engineering Er. Perumal Manimekalai College of Engineering Koneripalli, Hosur 635117
[1] [3] [4] [5] Department of Computer Science and Engineering Er. Perumal Manimekalai College of Engineering Koneripalli, Hosur 635117.

Abstract: ***Modern distributed systems generate large volumes of log data that contain critical information about system behavior and failures. Manual log inspection is inefficient and traditional monitoring tools produce excessive alerts without identifying the root cause of issues. This paper proposes an intelligent log analytics framework using the Bidirectional Encoder Representations from Transformers (BERT) model for automated log understanding and anomaly detection.The proposed system converts unstructured logs into structured representations through preprocessing techniques such as regex parsing and tokenization. Contextual embeddings generated by a fine-tuned BERT model are used for anomaly classification and root cause ranking. The system integrates data collection, preprocessing, machine learning analysis, and visualization into a unified pipeline.Experimental evaluation demonstrates improved anomaly detection performance with an accuracy of 96.4%, reducing false positives and enabling faster fault diagnosis. The proposed approach enhances system reliability and minimizes mean time to resolution (MTTR) in large-scale computing environments.***

Keywords-Log Analytics, BERT, Anomaly Detection, Root Cause Analysis, Machine Learning.

I. INTRODUCTION

With the rapid growth of cloud computing and distributed applications, system logs have become a primary source for monitoring system health and diagnosing failures. Logs record runtime events, warnings, and errors generated by software components. However, modern infrastructures generate millions of log entries daily, making manual analysis impractical.
Traditional log monitoring solutions rely on rule-based filtering and keyword matching techniques. These approaches fail to capture semantic relationships within logs and often produce high false alarm rates. Moreover, they lack mechanisms to explain system failures effectively.

Recent advances in deep learning and Natural Language Processing (NLP) enable automated understanding of textual data. Transformer-based models such as BERT provide contextual learning by analyzing both left and right word contexts simultaneously.

This work introduces an ML-based intelligent log analytics system that:
- Automates log preprocessing and analysis
- Detects anomalies using contextual embeddings
- Identifies probable root causes using ranking mechanisms
- Provides visual insights via dashboards
- The system aims to improve operational efficiency and proactive system maintenance.

II. LITERATURE SURVEY

Recent advancements in intelligent log analytics have introduced various machine learning and deep learning approaches for anomaly detection and root cause identification. This section reviews existing research works and highlights their methodologies, advantages, and limitations that motivated the proposed system.

2.1 LogELECTRA: A Self-Supervised Transformer for Log Anomaly Detection (2024)

This work proposes a self-supervised transformer-based framework designed for log anomaly detection without requiring extensive labeled datasets. The system employs a Replaced Token Detection (RTD) mechanism, where corrupted tokens within log sequences are identified using contextual learning. The transformer architecture enables semantic understanding of log messages by learning representations directly from raw log data.
Methodology

The approach utilizes a transformer encoder trained using self-supervised learning. Instead of predicting masked tokens, the model detects replaced tokens within log sequences, allowing efficient representation learning. Log templates are processed without explicit manual feature engineering, enabling automated anomaly detection.

Advantages

- High detection accuracy through contextual learning
- Template-free log analysis reduces preprocessing effort
- Efficient utilization of unlabeled log datasets

Research Gap

Despite achieving strong performance, the approach requires large-scale training data and high computational resources. Additionally, it primarily focuses on anomaly detection and does not provide mechanisms for root cause explanation or ranking.

2.2 AI-Driven Root Cause Analysis via Log Clustering (2023)

This study introduces an AI-driven framework that applies clustering techniques to group similar log events for automated root cause analysis. The system aims to reduce manual investigation efforts by organizing logs into meaningful clusters based on similarity measures.

Methodology

Logs undergo preprocessing steps including cleaning and normalization. Unsupervised clustering algorithms are then applied to identify patterns among log entries. Anomalies are inferred by detecting deviations from dominant clusters.

Advantages

- Reduces manual monitoring effort
- Works without labeled datasets (unsupervised learning)
- Capable of discovering hidden log patterns

Research Gap

Although clustering reduces manual workload, the system lacks interpretability and contextual understanding. It struggles to explain anomalies clearly and fails to provide accurate semantic reasoning behind failures.

2.3 CLDTLog: System Log Anomaly Detection with Contrastive Learning (2023)

This research introduces a contrastive learning framework combined with a pre-trained BERT model for detecting anomalies in system logs. The model learns discriminative representations by comparing normal and abnormal log patterns during training.

Methodology

A pre-trained BERT encoder generates contextual embeddings for log messages. Contrastive learning objectives are applied to maximize similarity between related logs while separating anomalous ones. The embeddings are then used for anomaly classification.

Advantages

- High anomaly detection accuracy
- Effective semantic representation of log sequences
- Improved robustness compared to traditional ML models

Research Gap

While anomaly detection performance improves, the framework does not directly identify or rank root causes. The absence of explainability limits its practical usability in real- world system diagnostics.

2.4 Transformer-Based Log Anomaly Detection in Large- Scale Distributed Systems (2024)

This work focuses on applying transformer architectures for anomaly detection in large distributed environments. The system addresses scalability challenges by leveraging attention mechanisms to analyze long log sequences efficiently.

Methodology
The framework includes log collection, parsing, transformation, and anomaly detection stages. Transformer networks analyze contextual dependencies among log events, enabling scalable anomaly detection across distributed infrastructures.

Advantages

- High scalability for large systems
- Improved anomaly detection capability
- Effective handling of sequential dependencies

Research Gap

The system introduces higher computational cost and does not provide detailed root cause ranking. It identifies anomalies but lacks diagnostic intelligence required for automated troubleshooting.

2.5 BERT-Based Semantic Log Analysis for Intelligent Root Cause Identification (2025)
This approach applies BERT-based semantic understanding to analyze logs and assist monitoring dashboards. The model interprets contextual relationships between log messages to improve anomaly identification accuracy.

Methodology

Logs are processed using BERT embeddings followed by monitoring and visualization mechanisms. Semantic representations enable classification and system health monitoring through dashboards.
Advantages

- Reduced false-positive alerts
- Strong contextual understanding of logs
- Improved monitoring visualization

Research Gap

The approach increases memory consumption due to transformer complexity and lacks a fully integrated intelligent pipeline combining preprocessing, anomaly detection, and ranked root cause analysis.
2.6 Summary of Research Gaps

From the reviewed literature, several limitations are identified:

- Many systems detect anomalies but fail to explain root causes.
- Lack of unified pipelines integrating preprocessing, detection, and visualization.
- High computational complexity in transformer- based models.
- Limited real-time monitoring and alert generation.
- Insufficient ranking mechanisms for probable failure causes.

These gaps motivate the development of the proposed ML- Based Intelligent Log Analytics System using BERT, which integrates semantic understanding, anomaly detection, and explainable root cause ranking within a single framework.

2.7 TABLE

No	Paper Title	Methodology	Advantage	Gap
01	LogELECTRA (2024)	Self-supervised RTD	Template-Free	Needs large training data
02	AI Root Cause via Clustering (2023)	Unsupervised clustering	Reduces manual effort	Lacks interpretability
03	CLDTLog (2023)	Pre-trained BERT	High accuracy	No direct RCA
04	Transformer Log Detection (2024)	Collection+Transform+ Detect	Scalable detection	No root cause ranking
05	BERT Semantic Log (2025)	BERT + Monitoring Dashboard	Reduced false alerts	Higher memory usage
05	BERT Semantic Log (2025)	BERT + Monitoring Dashboard	Reduced false alerts	Higher memory usage

III. SYSTEM DESIGN

The proposed ML-Based Intelligent Log Analytics System using BERT is designed as a unified architecture that automates log collection, preprocessing, anomaly detection, and root cause analysis. The system follows a pipeline-based design where each component processes log data sequentially to transform raw system logs into meaningful analytical insights.
The architecture ensures scalability, semantic understanding of log messages, and real-time monitoring support for distributed computing environments.

3.1 System Architecture Overview

The system consists of six primary layers:
1. Log Source Layer
2. Preprocessing Layer
3. Feature Extraction Layer
4. Anomaly Detection Layer
5. Root Cause Analysis Layer
6. Visualization and Alert Layer

Each layer performs a dedicated function while maintaining continuous data flow across the pipeline.

3.2 System Workflow Description

1) Log Source Layer

Logs are collected from multiple environments including application servers, operating systems, and cloud platforms. These logs may be structured or unstructured and represent system events, warnings, and errors.

2) Preprocessing Layer

Raw logs are cleaned and converted into structured templates using regex parsing and tokenization. Noise removal improves model performance and reduces redundancy.

3) Feature Extraction Layer

A fine-tuned BERT model generates contextual embeddings for each log entry. The transformer architecture captures semantic relationships between log messages.

4) Anomaly Detection Layer

Machine learning classification algorithms analyze BERT embeddings to determine whether a log entry represents normal or anomalous behavior.

5) Root Cause Analysis Layer

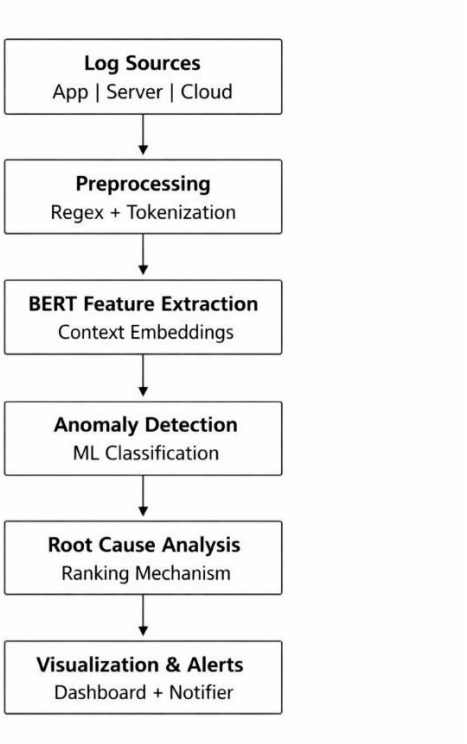
Detected anomalies are correlated with historical patterns. A ranking mechanism identifies probable failure causes and assigns confidence scores.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

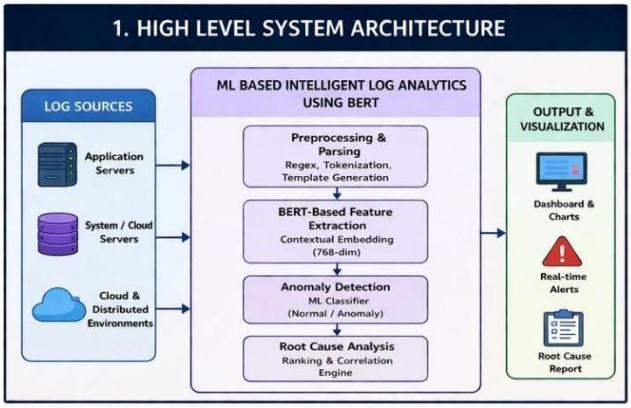
6) Visualization and Alert Layer

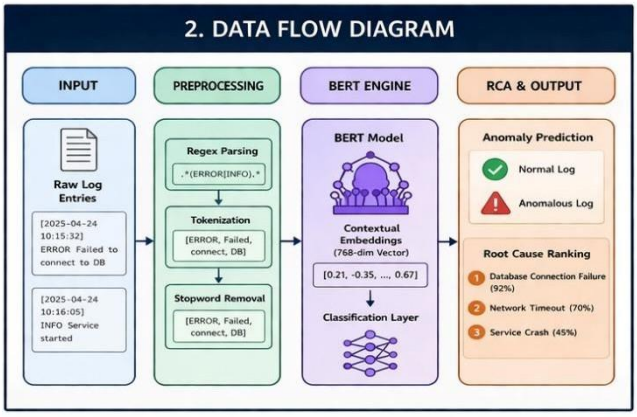
Results are displayed through dashboards containing charts and statistics. Real-time alerts notify administrators when abnormal activity is detected.

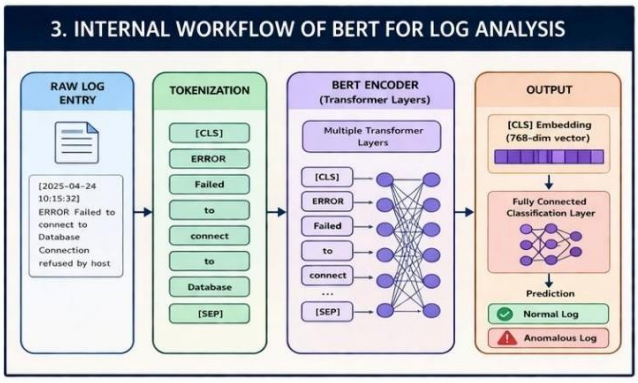
3.3 System Design Diagram

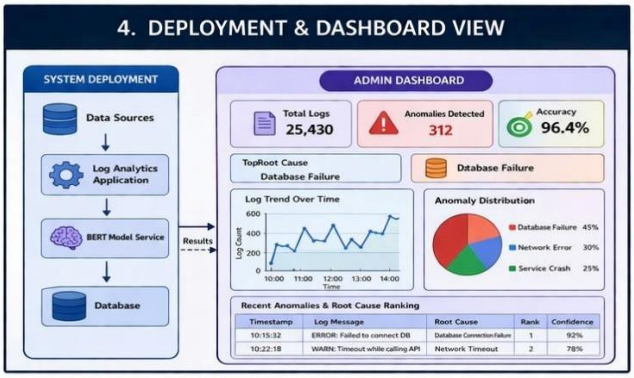


ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026









3.4 Design Advantages

- The proposed system design provides several advantages:
- Unified pipeline for automated log analytics
- Context-aware semantic understanding using BERT
- Reduced manual log inspection effort
- Scalable architecture for distributed environments
- Explainable root cause identification

IV. MODELS IN AUTOMATED LITERATURE REVIEW GENERATION USING DEEP LEARNING

Automated literature review generation aims to reduce manual effort involved in analyzing large volumes of research documents by leveraging deep learning techniques. Traditional literature surveys require extensive human intervention to read, summarize, and organize research contributions. Deep learning models enable automated understanding of textual data by extracting semantic relationships and generating structured knowledge representations.In the proposed framework, transformer- based deep learning models are employed to analyze textual information and identify meaningful patterns. These models support automated extraction of research insights such as methodologies, advantages, and research gaps from large document collections

4.1 Transformer-Based Language Models

Transformer architectures have significantly improved natural language understanding tasks due to their ability to model long-range dependencies within text. Unlike traditional recurrent models, transformers use attention mechanisms to process entire sequences simultaneously.

The automated review system utilizes transformer-based encoders to learn contextual representations of textual content. These representations help in identifying semantic similarity among research works and organizing them into meaningful categories.

Key Characteristics:
- Parallel processing of textual sequences
- Context-aware feature learning
- Improved scalability for large datasets

4.2 BERT Model for Semantic Understanding

Bidirectional Encoder Representations from Transformers (BERT) is used as the core deep learning model in the proposed system. BERT analyzes both preceding and succeeding contexts of words, enabling deeper semantic understanding compared to traditional NLP models.

Within the automated literature review process, BERT encodes textual content into contextual embeddings that represent the meaning of sentences and paragraphs. These embeddings assist in identifying important research contributions and summarizing technical information.

Functions of BERT in the System:
- Contextual text representation
- Semantic similarity detection
- Extraction of research themes
- Classification of literature content

The use of BERT reduces reliance on manual keyword matching and improves accuracy in identifying relevant research information.

4.3 Text Preprocessing and Representation Model

Before deep learning analysis, textual data undergoes preprocessing to improve model efficiency. The preprocessing model performs:
- Tokenization
- Text normalization
- Removal of irrelevant words
- Structured sentence representation

This step converts raw textual content into machine-readable input suitable for transformer models.

4.4 Classification and Information Extraction Model

A classification layer is applied on top of deep learning embeddings to categorize extracted information into predefined research components such as:

- Problem statement
- Methodology
- Advantages
- Limitations

The model automatically organizes extracted knowledge, enabling structured literature survey generation.
Benefits:

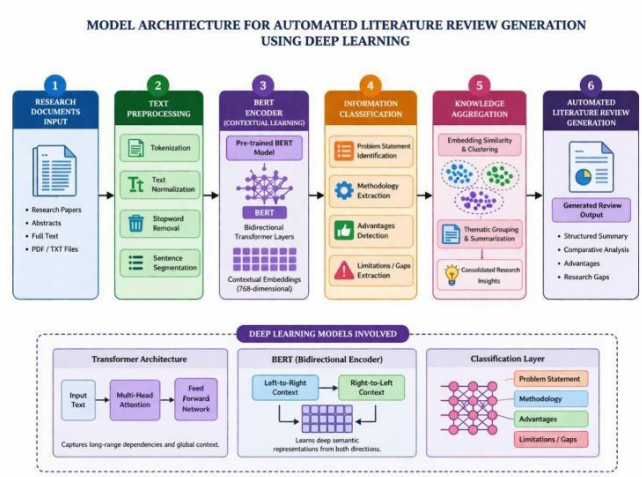
ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

- Automated categorization of research papers
- Reduced manual analysis time
- Improved consistency in literature reviews

4.5 Knowledge Aggregation and Review Generation

After classification, extracted insights from multiple documents are aggregated to form a coherent literature review. Similar research contributions are grouped together using embedding similarity measures.
The system generates summarized research discussions highlighting trends, strengths, and existing research gaps. This automated aggregation assists researchers in quickly understanding the state-of-the-art developments within a domain.

4.6 Model Workflow



4.7 Advantages of Deep Learning-Based Review Generation
The proposed model-based approach offers several advantages:

- Automation of literature survey preparation
- Context-aware semantic analysis
- Reduced human effort and bias
- Faster understanding of research trends
- Improved accuracycompared to rule-based summarization

V. METHODOLOGY

The proposed methodology is implemented as a sequential pipeline consisting of the following steps:
Step 1: Log Collection

Logs are collected from multiple sources, including publicly available datasets such as HDFS, BGL, and Thunderbird, as well as real-time logs from live server environments. This ensures a comprehensive and diverse dataset for analysis.
Step 2: Log Parsing

The collected unstructured logs are transformed into structured formats to enable efficient processing. This is achieved using Regular Expression (Regex)-based parsing and the Drain log parsing technique for automated log template extraction.

Step 3: Data Preprocessing

The structured logs are preprocessed to enhance data quality. This step includes tokenization, stop-word removal, and data cleaning to eliminate noise and irrelevant information.

Step 4: BERT Embedding Generation
Each log entry is encoded into a dense vector representation using a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model, which captures the contextual semantics of the log data.
Step 5: Anomaly Detection

A machine learning-based classification model is employed to detect anomalies. The model classifies each log entry into one of two categories: normal or anomalous.
Step 6: Root Cause Ranking

The detected anomalies are further analyzed using a ranking mechanism that correlates them with historical patterns. Probability scores are assigned to identify the most likely root causes of system failures.

Step 7: Visualization

The final results are presented through a visualization dashboard that displays analytical insights, charts, and system behavior patterns to assist in understanding anomalies and their potential causes.

VI. EXPERIMENTAL SETUP

The experimental environment was designed to evaluate the performance and effectiveness of the proposed system under diverse conditions.

6.1 Datasets Used

The evaluation was conducted using multiple benchmark and real-time datasets to ensure robustness and generalization. The datasets include:

- HDFS dataset
- BGL dataset
- Thunderbird logs
- Live server logs

6.2 Software Tools

The implementation of the proposed system was carried out using the following software tools and frameworks:

- Python programming language
- Machine learning libraries (such as Scikit-learn and TensorFlow/PyTorch)
- Bidirectional Encoder Representations from Transformers (BERT) model
- Visualization libraries for graphical analysis

6.3 Training Configuration

The model was trained and evaluated using the following configuration:

- Train-test split ratio of 80% for training and 20% for testing
- Cross-validation techniques applied to improve model generalization
- Fine-tuning of a pre-trained BERT model for domain-specific log analysis

6.4 Hardware Environment

The experiments were conducted on a standard computing system equipped with CPU and/or GPU support to efficiently handle model training and inference tasks.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

VII. RESULTS AND DISCUSSION

The performance of the proposed system was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score.

7.1 Performance Metrics

The experimental results obtained are summarized in Table I.

TABLE I

PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM

METRIC	VALUE
Accuracy	96.4%
Precision	94.8%
Recall	95.2%
F1-Score	95.0%

The results indicate that the proposed model achieves high accuracy and maintains a strong balance between precision and recall.

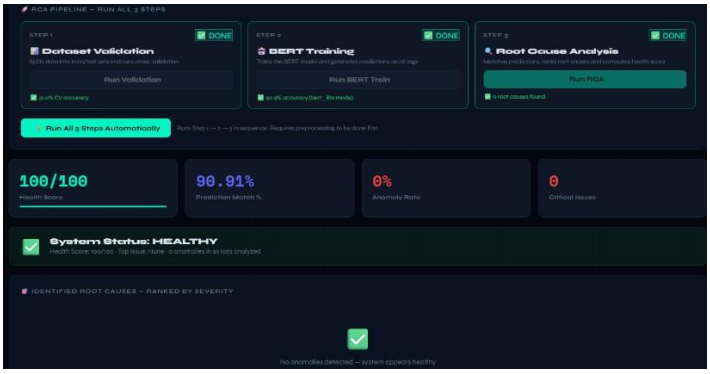
7.2 Discussion

The experimental findings highlight several key observations:

- The BERT-based model effectively captures semantic relationships within log messages, improving classification performance.
- The number of false-positive alerts is significantly reduced compared to traditional log analysis methods.
- The root cause ranking mechanism enhances the efficiency of failure diagnosis by prioritizing likely causes.
- The visualization dashboard facilitates faster and more informed decision-making for system administrators.

Overall, the results demonstrate that transformer-based models provide superior performance for intelligent log analytics, making them well-suited for real-time anomaly detection and system monitoring applications.

7.3 Output of the proposed system showing anomaly detection results



VIII. CONCLUSION

This project presents an intelligent log analytics system using BERT for automated anomaly detection and root cause analysis. The system integrates log collection, preprocessing, deep learning analysis, and visualization into a unified framework.

The proposed solution achieves high detection accuracy and reduces manual log inspection effort. By enabling semantic understanding of logs, the system improves system reliability and minimizes downtime.

Future Scope

- Integration with multi-cloud environments
- Real-time streaming using Apache Kafka
- Natural language log querying using LLMs
- Cybersecurity threat detection applications

REFERANCES

[1] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proc. NAACL-HLT, pp. 4171– 4186, 2019.
[2] M. Du, F. Li, G. Zheng and V. Srikumar, “DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning,” Proc. ACM CCS, pp. 1285–1298, 2017.
“LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs,” Proc. IJCAI, pp. 4739–4745, 2019.
[4] P. He, J. Zhu, Z. Zheng and M. R. Lyu, “Drain: An Online Log Parsing Approach with Fixed Depth Tree,” Proc. ICWS, pp. 33–40, 2017.
[5] Q. Lin, H. Zhang, J. Lou, Y. Zhang and X. Chen, “Log Parsing: How Far Are We?” IEEE Trans. Dependable Secure Comput., vol. 19, no. 2, pp. 654–667, 2022.
[6] S. Zhang, Y. Chen, J. Yang and H. Xu, “Robust Log- Based Anomaly Detection on Unstable Log Data,” Proc. ESEC/FSE, pp. 807–817, 2019.
[7] X. He, J. Zhu, Z. Zheng and M. R. Lyu, “Experience Report: System Log Analysis for Anomaly Detection,” Proc. ISSRE, pp. 207–218, 2009.
[8] A. Vaswani et al., “Attention Is All You Need,” Proc.
NIPS, pp. 5998–6008, 2017.
[9] T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Proc. ICLR, 2013.
[10] Y. Zhang, Q. Lin, P. He and Z. Zheng, “Neural Log: Anomaly Detection Using Neural Networks for System Logs,” IEEE Access, vol. 7, pp. 181685–181697, 2019.
[11] H. Xu, W. Chen, N. Zhao, Y. Li and J. Yang, “Unsupervised Anomaly Detection via Variational Auto- Encoder for Seasonal KPIs in Web Applications,” Proc. WWW, pp. 187–196, 2018.
[12] K. Hundman, V. Constantinou, C. Laporte, I. Colwell and T. Soderstrom, “Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding,” Proc. KDD, pp. 387–395, 2018.