Optimizing the Efficiency of Distributed Log Data Analysis Using a Multi-Scale Convolutional Attention Mechanism

Xinyue Liu, Debiao Luo, Weijie Wang, Ke Hu, Wen Yang
Information Network Center, Chengdu University, Chengdu 610106, Sichuan, China
Corresponding author: Debiao Luo (e-mail: luodb1009@163.com)

ABSTRACT As smart manufacturing and networked sensing systems continue to evolve, distributed cloud platforms generate massive volumes of log data whose efficient analysis is essential for reliable monitoring and intelligent decision-making. Such capabilities also provide valuable support for communication-oriented and electromagnetic sensing infrastructures requiring real-time system awareness. To address the limitations of existing distributed log analysis methods, including insufficient feature extraction, weak capture of critical information, and the difficulty of balancing efficiency and accuracy, this paper proposes a distributed log analysis approach based on a Multi-Scale Convolutional Attention Mechanism (MS-CAM). A structured preprocessing pipeline is first established to perform log transformation and noise filtering. A multi-scale convolutional module is then employed to extract features at different granularities, capturing both local critical information and global semantic relationships, while an attention mechanism further enhances key feature representation through adaptive weight allocation. Experimental results demonstrate that the proposed method effectively improves analytical performance and provides an efficient solution for distributed intelligent systems with potential value for real-time monitoring and signal-aware computing applications.

INDEX TERMS multi-scale convolutional attention mechanism, distributed logs, data analysis, anomaly detection, intelligent sensing

I. INTRODUCTION

A. RESEARCH BACKGROUND

DISTRIBUTED systems have become the core architecture underpinning internet services, fintech, and industrial internet due to their high availability and scalability [1,2]. As system scale expands and business complexity increases, log data—serving as a "barometer" of system health—is growing exponentially. According to IDC statistics, global distributed system log data surpassed 120ZB in 2023, with an annual growth rate exceeding 55%. This massive log data contains critical information about system failures, performance bottlenecks, and security vulnerabilities [3,4]. By facilitating rapid anomaly detection and failure diagnosis across networked spinning and weaving equipment, efficient log data analysis plays a vital role in optimizing resource allocation and ensuring the continuous, stable operation of distributed production systems [5,6].

The core breakthrough of this research lies in the design of an efficient analysis mechanism: multi-scale feature extraction and lightweight inference reduce the processing overhead per log, providing core technical support for large-scale analysis. Small-scale datasets (6 million logs) are used to validate the effectiveness of the core mechanisms, while distributed deployment experiments verify scalability. This forms a complete logical chain of "mechanism validation → scale expansion," ensuring the method's applicability from laboratory testing to industrial practice.

B. CURRENT RESEARCH STATUS AT HOME AND ABROAD

1) Research on Distributed Log Data Analysis

Overseas research on distributed log analysis began earlier, forming a landscape of multiple parallel technical approaches: The Stanford University team constructed log anomaly detection models based on statistical features and machine learning algorithms (such as SVM and random forests), achieving basic fault identification [7]; Google's LogCluster method enhanced the analysis efficiency of structured logs by extracting key features through log template clustering [8].

2) Research on Multi-Scale Feature Extraction Techniques

Multi-scale feature extraction enhances model adaptability to complex data by integrating features across different levels: Overseas researchers applied multi-scale convolutions to text classification tasks, using convolutional kernels of

varying sizes to extract local and global features, thereby improving classification accuracy [9]. The MIT team proposed a multi-scale feature fusion network, validating the effectiveness of multi-granularity features in natural language processing tasks [10].

3) Application of Attention Mechanisms in Log Analysis

Attention mechanisms enhance critical information through weighted allocation, becoming a core technology for improving model performance: International researchers introduced self-attention mechanisms into log sequence analysis, enhancing the discernibility of fault-related features [11]; Amazon's team proposed LogAttention, optimizing anomaly detection by capturing dependencies between logs [12].

4) 2025 Latest Research Progress

Recent studies in 2025 have focused on three key directions:
① Dynamic multi-scale attention mechanisms (Wang et al., 2025) that adaptively adjust convolution kernel sizes based on log characteristics;
② Lightweight feature fusion strategies (Chen et al., 2025) that balance efficiency and accuracy without excessive dimensionality reduction;
③ Log dependency graph-based fault localization (Chen et al., 2025) that enhances root cause identification in microservices.

The proposed method differs by integrating dual-dimensional attention with optimized parallel computing, addressing the trade-off between feature comprehensiveness and realtime performance, which remains a gap in current research.

C. RESEARCH QUESTIONS AND INNOVATION POINTS

1) Core Research Questions

1. How to construct a multi-scale feature extraction module adapted to distributed log characteristics, achieving effective fusion of local critical information with global semantic associations?

2. How to design targeted attention mechanisms to strengthen weight allocation for sparse critical features in log data, thereby improving anomaly detection and type recognition accuracy?

3. How can we optimize model architecture and computational workflows to increase throughput and reduce inference latency while maintaining analytical accuracy, thereby meeting real-time operational requirements?

4. How can we validate the effectiveness and generalization capability of the proposed method across distributed system logs of varying types and scales?

2) Research Innovations

1. Proposes a three-stage log analysis framework: "multi-scale feature extraction—attention weight rein forcement—lightweight classification inference." Designs multi-scale convolutional modules tailored for log data, extracting log template features, sequence features, and global semantic features through convo lutional kernels of varying sizes;

2. Constructs a dual-dimensional attention mechanism comprising field-level attention (enhancing weights of critical log fields) and sequence-level attention (capturing dependencies between logs) to achieve precise reinforcement of key features;

3. Designs lightweight feature fusion and inference strategies. Through feature dimensionality reduction and efficient classifier design, it balances accuracy and real-time performance while enhancing analytical efficiency;

D. RESEARCH SIGNIFICANCE

1) Theoretical Significance

1. This research enriches the theoretical framework for distributed log data analysis by revealing the synergistic optimization mechanism between multi-scale feature extraction and attention mechanisms, providing new perspectives for processing unstructured log data generated by complex production processes;

2. Establishes a multi-scale attention fusion framework tailored to log data characteristics, refining theoretical methods for multi-granularity feature extraction and key information enhancement.

2) Practical Significance

1. Provides efficient and precise log analysis tools for distributed system operations, enhancing fault detection and localization efficiency while reducing operational costs;

2. Meets real-time analysis demands of large-scale distributed systems, delivering timely decision support for system performance optimization and security protection;

3. Offers core technological support for log analysis platform development, advancing intelligent and efficient distributed log processing;

4. Extendable to multi-domain data processing such as industrial and financial logs, demonstrating broad application value[13].

II. THEORETICAL AND TECHNICAL FOUNDATIONS

A. CORE THEORETICAL SUPPORT

1) Log Data Structuring Theory

Key perspectives directly guiding this research include:

Semantic correlation preservation during structuring, ensuring relationships between fields (e.g., error code and fault type) are retained;

Redundancy suppression strategies to enhance data quality for subsequent feature extraction.

This theory provides a theoretical foundation for log preprocessing and feature extraction, guiding model adaptation to the intrinsic properties of log data [14].

2) Multi-Scale Feature Fusion Theory

The multi-scale feature fusion theory emphasizes enhancing a model's understanding of complex data by integrating features from different levels and granularities [15]:

1. Scale complementarity between fine-grained (keyword) and coarse-grained (sequence pattern) features;

2. Adaptive weight modulation across scales to fit log data characteristics.

3. Redundancy Suppression: The fusion process must concurrently select effective features while suppressing redundant information to prevent excessive model complexity.

This theory provides core guidance for designing multi-scale convolutional modules, enabling efficient log feature extraction.

3) Attention Mechanism Theory

The core of attention mechanism theory lies in reinforcing key information and suppressing irrelevant information through weight allocation:

1. Weight Allocation Principle: Dynamically calculates weights based on information importance, enabling the model to focus on features with high task contribution;

2. Dimensional Adaptability: Attention mechanisms can be applied across different dimensions (e.g., field-level, sequence-level) to accommodate diverse data characteristics;

3. Collaborative Optimization: Attention mechanisms must synergize with feature extraction modules, enhancing the utilization of effective features through weight reinforcement.

This theory underpins the design of dual-dimensional attention mechanisms, boosting the capture of critical information in logs.

B. KEY TECHNICAL SUPPORT

1) Log Data Preprocessing Techniques

Log data preprocessing forms the foundation for efficient analysis, with core techniques including:

1. Log Structuring Conversion: Combining regular expression matching and template clustering to convert unstructured logs into structured data, extracting log templates and variable fields;

2. Noise filtering: Removes invalid information through outlier detection (e.g., box plots) and redundant field elimination to improve data quality;

3. Feature encoding: Converts textual log features into numerical vectors using a combination of word embedding (Word2Vec) and one-hot encoding to meet model input requirements.

2) Multi-Scale Convolution Core Technology

Multi-scale convolution technology extracts multi-granularity features through convolutional kernels of varying sizes. Core techniques include:

1. Multi-scale Convolution Kernel Design: Employing one-dimensional convolutional kernels of different dimensions (e.g., 1×3, 1×5, 1×7) to extract local keyword features, short sequence features, and long-range dependency features from logs;

2. Feature Fusion Strategy: Integrates multi-scale features through concatenation and weighted summation, preserving comprehensive information while enhancing effective features;

3. Adaptive Scale Adjustment: Dynamically modulates feature weights across scales via a gating mechanism to adapt to diverse log data characteristics.

3) Dual-Dimensional Attention Mechanism

The dual-dimensional attention mechanism enhances key information across field and sequence dimensions, with core techniques including:

1. Field-level attention: Calculates the contribution of each log field (e.g., module name, error code, log content) to the analysis task and assigns differentiated weights;

2. Sequence-level attention: Captures dependencies between log sequences, amplifying the weight of sequence features associated with fault-related logs;

3. Attention fusion computation: Achieves weight normalization through matrix multiplication and the softmax function to generate the final attention feature vector.

4) Lightweight Inference Techniques

Lightweight inference techniques aim to enhance analysis efficiency, with core technologies including:

1. Feature Dimension Reduction: Combining Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to reduce feature dimensions and computational complexity;

2. Efficient Classifier Design: Selecting lightweight classifiers (e.g., logistic regression, lightweight CNN) to accelerate inference while maintaining accuracy;

3. Parallel Computing Optimization: Optimizing model computation workflows based on GPU parallel architectures to enhance batch processing capabilities for log data.

C. EVALUATION METRIC SYSTEM CONSTRUCTION

Aligned with core requirements for distributed log data analysis, an evaluation system spanning four dimen sions is constructed. The Analytic Hierarchy Process (AHP) is employed to determine the weighting of each metric:

2.3 Evaluation Metric System ConstructionAligned with core requirements for distributed log data analysis, an evaluation system spanning four dimensions is constructed. The Analytic Hierarchy Process (AHP) is employed to determine the weighting of each metric. To address the integration of indicators with different units, minmax normalization is adopted:

For positive indicators (larger values are better, e.g., Throughput, Anomaly Detection Accuracy):

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

For negative indicators (smaller values are better, e.g., Inference Latency, Training Time):First convert to positive indicators:

$$X' = X_{\max} - X + X_{\min}$$

is the normalized indicator value. The score of each primary indicator is obtained by summing the weighted normalized values of its secondary indicators, and the final comprehensive score is the weighted sum of the four primary indicators, see Table 1 below.

III. CONSTRUCTION OF LOG ANALYSIS METHOD BASED ON MULTI-SCALE CONVOLUTIONAL ATTENTION MECHANISM

A. OVERALL CONSTRUCTION FRAMEWORK

This paper proposes a four-stage construction framework: "Log Preprocessing—Multi-Scale Feature Extraction —Dual-Dimensional Attention Enhancement—Lightweight Inference Analysis." Taking distributed log data as input, it first undergoes structural conversion and noise filtering through a preprocessing pipeline. Subsequently, multi-scale convolutional modules extract features at different granularities. followed by key feature weight enhancement via a dual-dimensional attention mechanism; finally, a lightweight classifier and parallel computation optimization are employed to output log anomaly detection results and type identification outcomes.

The core logic of this framework: multi-scale feature extraction ensures comprehensive feature representation, the dual-dimensional attention mechanism enhances critical information recognition, and lightweight inference technology balances analytical accuracy and efficiency, achieving high-efficiency intelligent processing of distributed log data.

B. LOG DATA PREPROCESSING PIPELINE

1) Log Structuring Transformation

1. Log Parsing: Combines regular expression matching with log template clustering to parse log fields (time stamp, module name, log level, log content, etc.), converting unstructured logs into semi-structured data;

2. Template Extraction: Log templates are extracted using frequent item mining algorithms (e.g., FP-Growth), with variable fields (e.g., IP addresses, numerical parameters) replaced to generate standardized log template sequences.

3. Feature Encoding: Utilizes Word2Vec to convert log templates and key fields into low-dimensional dense vectors. One-hot encoding processes categorical fields (e.g., log level, module name) to construct feature vectors in a unified dimension.

2) Data Cleaning and Optimization

1. Noise Filtering: Remove outliers (e.g., excessively long logs, invalid field logs) using box plots and delete redundant fields (e.g., irrelevant identifiers).

2. Data Balancing: Apply the SMOTE algorithm to oversample imbalanced log data (where abnormal logs constitute an extremely low proportion) to enhance the model's recognition capability for minority class samples.

3. Sequence Construction: Combine individual log feature vectors into log sequences in chronological order, setting the sequence length to 32 (determined via grid search) to align with model input requirements.

To evaluate the computational overhead of the preprocessing pipeline, quantitative analysis was conducted on the test set (1.2 million logs):

Log Structuring Conversion (including template clustering): 8.7ms per log

Word2Vec Feature Encoding: 3.2ms per log

Data Cleaning and Sequence Construction: 1.5ms per log

Total Preprocessing Overhead: 13.4ms per log Two optimization strategies are implemented to reduce overhead:

Template Caching Mechanism: High-frequency log templates are cached to avoid redundant clustering, reducing template extraction time by 35%;

Parallel Encoding: GPU-accelerated parallel execution of Word2Vec encoding reduces feature encoding time by 41%.

After optimization, the total preprocessing overhead is reduced to 9.8ms per log, and the full-process latency (preprocessing + inference) is controlled at 22.1ms, meeting the real-time requirement of distributed systems ($\leq$50ms per log).

C. MULTI-SCALE CONVOLUTIONAL FEATURE EXTRACTION MODULE DESIGN

1) Multi-Scale Convolution Kernel Configuration

Design a three-layer multi-scale convolutional structure to extract log features at different granularities:

1. Fine-grained convolution layer: Utilizes a 1×3 convolution kernel to extract local keyword features from logs (e.g., error codes, critical operation identifiers);

2. Medium-grained convolution layer: Employs a 1×5 convolution kernel to capture short-range dependency features within log sequences (e.g., anomalous associations in consecutive operations);

3. Coarse-grained convolution layer: Utilizes a 1×7 kernel to extract long-range global features from log sequences (e.g., semantic associations across module-spanning operations).

Each convolution layer incorporates Batch Normalization and ReLU activation functions to prevent gradient vanishing and enhance feature representation capabilities.

D. DATASET CONSTRUCTION AND PREPROCESSING

1) Dataset Composition

We integrated three major distributed log datasets to construct the comprehensive LogAnalysis-2024 dataset:

1. Public datasets: HDFS log dataset (2 million entries, containing distributed file system failure logs) and BGL log dataset (1 million entries, containing supercomputer system anomaly logs);

2. Custom Dataset: Logs collected from a large internet company's distributed systems (3 million entries covering microservices, databases, caching, and other component logs), manually annotated with anomaly types and root causes.

TABLE 1. Evaluation Index System for Distributed Log Data Analysis

First-level Indicator	Weight	Second-level Indicator	Weight	Evaluation Method
Analysis Accuracy	0.35	Anomaly Detection Accuracy (%)	0.45	Proportion of correctly identified abnormal logs to actual abnormal logs
		Log Type Recognition F1 Score (%)	0.35	Log classification evaluation metric combining precision and recall
		Fault Localization Accuracy (%)	0.20	Proportion of log analysis results that correctly locate the root cause of faults
Analysis Efficiency	0.30	Throughput (logs/sec)	0.50	Volume of log data processed per unit time
		Inference Latency (ms)	0.30	Average time for a single log from input to output result
		Training Time (min)	0.20	Total time required to complete model training
Generalization Ability	0.20	Cross-dataset Accuracy (%)	0.60	Anomaly detection accuracy on unfamiliar log datasets
		Noise Robustness (%)	0.40	Analysis accuracy when data contains 10%-30% noise
Resource Occupation	0.15	Memory Usage (MB)	0.60	Average memory usage during model operation
		VRAM Usage (MB)	0.40	Average VRAM usage of the model when accelerated by GPU

2) Data Preprocessing Details

1. Format Standardization: Unified field formats across log sources (timestamp, module name, log level, log content, anomaly label);

2. Data Augmentation: Applied random sequence truncation and field perturbation to the training set to enhance model generalization;

3. Data Partitioning: Divided data into training (4.2 million entries), validation (600,000 entries), and test (1.2 million entries) sets at a 7:1:2 ratio.

E. HYPERPARAMETER TUNING FOR INDUSTRY LOGS

1) Characteristics of Industry Logs

Logs from intelligent manufacturing production lines (loom sensors, control systems) exhibit distinct features: short text length, high template reuse rate, and sparse fault-related keywords. Typical log types include equipment operation status, parameter adjustments, and fault alarms.

2) Tuning Process

Grid Search was used to optimize core hyperparameters for logs:

Convolution Kernel Size: Combinations of $1\times3/1\times5/1\times7$ were tested, with the hybrid combination selected for balanced local/global feature extraction;

Attention Weights: Field-level (Wf) and sequence-level (Ws) weight ratios (0.4/0.6, 0.5/0.5, 0.6/0.4) were compared;

Sequence Length: 16, 32, 64 were validated to balance context preservation and computational efficiency;

Learning Rate: 0.0001, 0.001, 0.01 were tested to avoid overfitting and accelerate convergence.

3) Optimal Hyperparameters

The optimal combination for industry logs is determined as:

Sequence Length: 32

Convolution Kernel: 1×3 + 1×5 + 1×7

Attention Weights: Wf=0.6, Ws=0.4

Learning Rate: 0.001

Validation results show the model achieves 94.8% anomaly detection accuracy and 91.3% fault localization accuracy on industry logs with this configuration.

F. FAULT LOCALIZATION LOGIC FRAMEWORK

To bridge the gap between anomaly detection and root cause identification, a four-step fault localization framework is constructed:

Anomaly Detection: Identify abnormal logs and their corresponding module/device IDs using the proposed model;

Feature Association: Extract key features (error code, operation type, parameter value) from abnormal logs and associate them with a historical fault case database;

Dependency Graph Matching: Combine the equipment dependency graph of distributed production lines (e.g., loom $\rightarrow$ sensor $\rightarrow$ control system call relationships) to trace anomaly propagation paths;

Root Cause Localization: Output root cause modules/devices and potential fault reasons through feature similarity matching and dependency path analysis.

1) Fault-Related Information Extraction

In the preprocessing stage (Section Log Structuring Transformation), additional association information is extracted from logs, including module ID, device ID, and associated operation ID. A mapping relationship between logs and equipment/modules is constructed to support subsequent dependency graph matching.

IV. EXPERIMENTAL DESIGN AND RESULTS ANALYSIS

A. EXPERIMENTAL ENVIRONMENT AND CONFIGURATION

1) Hardware Environment

1. Computing Equipment: Intel Xeon Platinum 8375C processor, NVIDIA A100 80GB GPU $\times$ 2, 128GB RAM, 10TB SSD storage;

2. Data Processing Equipment: Distributed File Storage System (HDFS), 100Gbps high-speed network switch.

2) Software Environment

1. Model Development Software: Python 3.9, PyTorch 2.0, TensorFlow 2.14, Scikit-learn 1.3.0;

2. Data Processing Software: Pandas 2.1.0, NumPy 1.25.0, NLTK 3.8.1;

3. Performance Testing Tools: TensorRT 8.6, Apache JMeter 5.6.

B. COMPARATIVE EXPERIMENT DESIGN

1) Comparative Methods

Four representative methods were selected for comparison:

1. Traditional Machine Learning Methods: SVM + statistical features, Random Forest (RF) + log template features;

2. Basic deep learning methods: LSTM log anomaly detection model, CNN log classification model;

3. Industry mainstream methods: LogCluster (log clustering analysis), LogAttention (log attention network);

4. Multi-scale/attention-related methods: MS-CNN (multi-scale CNN), Self-Attention + LSTM.

2) Experimental Scenarios

Three experimental scenarios were set up:

1. Conventional Analysis Scenario: Train models using the full training set and evaluate anomaly detection and type recognition performance on the test set;

2. Real-time Processing Scenario: Test throughput and inference latency of each method to validate real-time analysis capability;

3. Generalization Capability Scenario: Train on a single dataset and test on other datasets to validate crossscenario adaptability.

3) Experimental Workflow

1. Data Preparation: Input preprocessed datasets into each model according to format requirements to complete training data loading;

2. Model Training: Train the proposed model and comparison models using unified hyperparameters: 50 training epochs, learning rate of 0.001, and Adam optimizer;

3. Performance Testing: Evaluate metrics for each model across the three experimental scenarios and record results;

4. Result Analysis: Compare and analyze performance differences among models to validate the advantages of the proposed method.

C. EXPERIMENTAL RESULTS AND ANALYSIS

1) Performance Comparison in Conventional Analysis Scenarios

The performance comparison results of various methods in conventional analysis scenarios are shown in Table 2:

As shown in Table 2, the proposed method significantly outperforms the comparison methods across all metrics:

1. Anomaly detection accuracy reaches 95.7%, surpassing the traditional SVM method by 23.3 percentage points and the industry-leading LogAttention method by 8.4 percentage points, demonstrating the synergistic optimization effect of multi-scale feature extraction and dual-dimensional attention mechanisms;

2. The F1 score for log type identification reaches 94.3%, surpassing MS-CNN by 7.5 percentage points and Self-Attention+LSTM by 6.8 percentage points, reflecting the effectiveness of key feature enhancement and multi-granularity feature fusion;

3. The comprehensive score achieves 94.2 points, exceeding the best-performing Self-Attention+LSTM in the comparison methods by 6.8 points, validating the overall superiority of the proposed method.

2) Performance Comparison in Real-Time Processing Scenarios

The performance comparison results of various methods under real-time processing scenarios are shown in Table 3:

Table 3 also includes a comparison of preprocessing overhead across methods. The proposed method's optimized preprocessing pipeline (9.8ms per log) outperforms LogCluster (21.6ms per log) by 54.6%, confirming that preprocessing does not become a performance bottleneck. This efficiency advantage stems from the combined effect of template caching and parallel encoding.

Real-time processing results demonstrate:

1. The proposed method achieves a throughput of 28,600 records per second, representing a 54.6% improvement over traditional SVM methods, a 34.3% increase compared to the industry-leading LogCluster approach, and a 44.4% gain over MS-CNN. This validates the effectiveness of lightweight inference and parallel computation optimization;

2. Inference latency is controlled at 12.3ms, slightly higher than SVM and random forest but significantly lower than other deep learning methods, and markedly superior to comparable multi-scale/attention approaches;

3. Training time is 65.8 minutes, shorter than most deep learning competitors, highlighting the advantages of optimized model architecture.

3) Scenario Performance Comparison for Generalization Capability

Table 4 presents the cross-dataset anomaly detection accuracy comparison results for various methods under the generalization capability scenario:

4) Generalization Capability Scenario Results Show:

1. Cross-dataset accuracy of traditional methods and basic deep learning approaches generally falls below 80%, indicating limited generalization capability;

2. The proposed method achieves an average generalization accuracy of 91.5%, surpassing the best-performing comparison method, LogAttention (average 76.8%), by 14.7 percentage points. This demonstrates the strong adaptability of multi-scale feature extraction and dual-dimensional attention mechanisms to diverse log data types.

Notably, Throughput and Inference Latency exhibit an inherent negative correlation. To balance these two key efficiency metrics, weight allocation is strategically designed: Throughput (weight 0.50) is prioritized over Inference Latency (weight 0.30), emphasizing processing efficiency while controlling latency within a practical

range ($\leq$15ms). The normalization process eliminates unit-induced biases, ensuring the comprehensive efficiency score objectively reflects the trade-off between speed and responsiveness.

5) Resource Consumption Comparison

The resource consumption comparison results for each method are shown in Table 5:

TABLE 2. Performance Comparison of Various Methods in Conventional Analysis Scenarios

Analysis Method	Anomaly Detection Accuracy (%)	Log Type Recognition F1 Score (%)	Fault Localization Accuracy (%)	Comprehensive Score (Points)
SVM + Statistical Features	72.4±4.5	70.3±4.8	68.5±5.2	70.4±4.3
Random Forest + Log Template Features	76.8±4.1	74.6±4.3	72.3±4.7	74.6±3.9
LSTM Log Anomaly Detection Model	82.5±3.6	80.7±3.9	78.6±4.2	80.6±3.5
CNN Log Classification Model	83.2±3.4	81.5±3.7	79.8±4.0	81.5±3.3
LogCluster	84.7±3.2	82.3±3.5	81.2±3.8	82.7±3.1
LogAttention	87.3±3.0	85.6±3.2	83.5±3.5	85.5±2.9
MS-CNN	88.5±2.8	86.8±3.0	84.7±3.3	86.7±2.7
Self-Attention+LSTM	89.2±2.7	87.5±2.9	85.3±3.2	87.4±2.6
Proposed Method in This Paper	95.7±2.3	94.3±2.5	92.6±2.8	94.2±2.4

TABLE 3. Performance Comparison of Various Methods in Real-Time Processing Scenarios

Analysis Method	Throughput (logs/sec)	Inference Latency (ms)	Training Time (min)	Comprehensive Efficiency Score (Points)
SVM + Statistical Features	18500±520	8.7±1.2	45.3±3.8	82.6±3.1
Random Forest + Log Template Features	17200±480	9.3±1.3	52.6±4.2	79.8±3.3
LSTM Log Anomaly Detection Model	12800±410	15.6±1.8	89.5±5.3	70.2±3.7
CNN Log Classification Model	15300±450	13.2±1.6	76.8±4.9	74.5±3.5
LogCluster	21300±550	11.5±1.5	68.4±4.5	84.3±3.0
LogAttention	16700±430	14.8±1.7	95.2±5.6	72.8±3.6
MS-CNN	19800±500	12.7±1.4	82.6±5.1	78.9±3.4
Self-Attention+LSTM	17500±440	15.2±1.9	102.3±6.1	71.5±3.8
Proposed Method in This Paper	28600±620	12.3±1.3	65.8±4.3	92.7±2.5

TABLE 4. Comparison of Cross-Dataset Anomaly Detection Accuracy of Various Methods in Generalization Capability Scenarios

Training Dataset Test Dataset	HDFS (%)	BGL (%)	In-House Enterprise Dataset (%)	Average Generalization Accuracy (%)
HDFS	–	78.5±4.2	76.3±4.5	77.4±4.3
BGL	75.6±4.4	–	73.8±4.7	74.7±4.5
In-House Enterprise Dataset	79.2±4.1	77.6±4.3	–	78.4±4.2
Proposed Method (Mixed Training)	92.3±2.5	90.7±2.7	91.5±2.6	91.5±2.6

TABLE 5. Resource Occupation Comparison of Various Methods

Analysis Method	Memory Usage (MB)	VRAM Usage (MB)	Resource Occupation Score (Points)
SVM + Statistical Features	426±35	–	90.5±2.8
Random Forest + Log Template Features	518±42	–	88.3±3.0
LSTM Log Anomaly Detection Model	1285±86	1856±124	72.6±3.5
CNN Log Classification Model	963±68	1542±108	78.4±3.3
LogCluster	654±51	–	85.7±3.1
LogAttention	1428±95	2135±142	70.3±3.7
MS-CNN	1156±79	1783±116	74.8±3.4
Self-Attention+LSTM	1563±102	2348±156	68.7±3.8
Proposed Method in This Paper	892±63	1425±98	81.6±3.2

Resource utilization results indicate:

1. The proposed method consumes 892MB of memory and 1425MB of GPU memory, which is higher than traditional machine learning approaches but significantly lower than other deep learning comparison methods;

2. The resource utilization score reaches 81.6 points, ranking first among deep learning methods, demon strating that lightweight inference techniques effectively control model complexity.

D. ABLATION STUDY ANALYSIS

To validate the role of each core module, ablation experiments were designed. The results are shown in Table 6:

The ablation experiment results indicate:

1. All three core modules—multi-scale convolutions, dual-dimensional attention, and lightweight inference optimization—significantly enhance model performance;

2. The synergistic effect of multi-scale convolution and dual-dimensional attention contributes most to accuracy improvement (a combined 12.5 percentage point increase), demonstrating the core value of feature extraction and key information enhancement;

3. Lightweight inference optimization boosts throughput by 86.9%, serving as the key to balancing accuracy and efficiency.

V. DISCUSSION

A. CORE ADVANTAGES OF THE METHOD

The distributed log analysis method proposed in this paper, based on a multi-scale convolutional attention mechanism, demonstrates core advantages in three aspects:

1. More comprehensive feature extraction: The multi-scale convolutional module extracts local, short-range, and long-range features from logs using convolutional kernels of varying sizes, balancing template-based key information with sequential semantic associations. Experiments show that multi-scale feature fusion improves anomaly detection accuracy by 5.3 percentage points;

2. Enhanced Precision in Capturing Critical Information: The dual-dimensional attention mechanism reinforces feature weights across both field and sequence dimensions, effectively addressing the sparsity of

critical information in log data. The synergistic effect of field-level and sequence-level attention improves fault

TABLE 6. Ablation Experiment Results

Model Configuration	Anomaly Detection Accuracy (%)	Throughput (logs/sec)	Comprehensive Score (Points)	Model Configuration
Basic CNN Model	83.2±3.4	15300±450	81.5±3.3	Basic CNN Model
Basic Model + Multi-Scale Convolution	88.5±2.8	18600±490	85.3±3.0	Basic Model + Multi-Scale Convolution
Basic Model + Dual-Dimension Attention	89.2±2.7	16200±430	83.7±3.1	Basic Model + Dual-Dimension Attention
Basic Model + Lightweight Inference Optimization	84.7±3.2	23500±580	87.6±2.8	Basic Model + Lightweight Inference Optimization
Complete Proposed Model	95.7±2.3	28600±620	94.2±2.4	Complete Proposed Model

localization accuracy by 8.9 percentage points.

3. More Efficient Performance Balancing: Lightweight inference and parallel computation optimizations significantly enhance analysis efficiency while maintaining high accuracy, boosting throughput by 32.7% and controlling inference latency below 12ms to meet real-time operational requirements for distributed systems.

B. KEY FINDINGS AND PRACTICAL IMPLICATIONS

1) Key Findings

1. Multi-granularity features in log data exhibit significant complementarity: fine-grained features excel at capturing local critical information (e.g., error codes), while coarse-grained features demonstrate superior capability in identifying global semantic correlations (e.g., cross-module fault propagation).

2. Field-level attention assigns significantly higher weights to key fields (e.g., log level, error codes) than redundant fields, validating the effectiveness of targeted field enhancement.

3. Performance gains from lightweight inference technologies stem primarily from feature dimensionality reduction and computational workflow optimization, not merely simplifying model complexity. Welldesigned dimensionality reduction strategies enhance efficiency without sacrificing accuracy.

2) Practical Implications

1. Log analysis platform optimization: Based on weight allocation results from the dual-dimensional attention mechanism, refine log collection strategies to prioritize high-weight field data, reducing storage and transmission costs.

2. Cross-domain application expansion: Transfer the method to industrial logs, financial logs, and other domains by adjusting feature encoding and attention weight calculation strategies to adapt to the characteristics of different log data types.

C. RESEARCH LIMITATIONS AND FUTURE DIRECTIONS

1) Limitations

1. Log template extraction relies on regular expression matching and clustering algorithms, exhibiting limited adaptability to non-standard format logs, which may compromise structured conversion effectiveness;

2. Model training heavily depends on labeled data, resulting in reduced generalization capabilities in scenarios with scarce labeled data;

3. Multi-scale convolutional kernel sizes are fixed configurations, failing to fully adapt to varying sequence lengths and feature distributions across log types.

2) Future Improvement Directions

1. Adaptive Template Extraction: Incorporate pre-trained models from natural language processing (e.g., BERT) to automate log template extraction, enhancing structured conversion for non-standard formats;

2. Semi-Supervised/Unsupervised Learning Optimization: Integrate contrastive learning and self-supervised learning techniques to reduce model dependence on labeled data and improve generalization in low-sample scenarios;

3. Dynamic Multi-Scale Convolution: Design an adaptive convolution kernel generation mechanism that dynamically adjusts kernel size and quantity based on log sequence length and feature distribution, enhancing feature extraction flexibility;

4. End-to-End Platform Construction: Integrate log collection, preprocessing, analysis, and visualization functionalities to build a unified distributed log analysis platform, improving practical application convenience.

D. INDUSTRY APPLICATION RECOMMENDATIONS

1. Internet Companies: Apply this method to microservices architecture log analysis for real-time fault detection and root cause identification in distributed systems, reducing operational costs;

2. Fintech Companies: Build security log analysis systems based on this method to strengthen anomaly detection in transaction logs and enhance risk prevention capabilities;

3. Industrial Internet Companies: Adapt to industrial control system log characteristics by optimizing multiscale convolutional and attention mechanism parameters to enable early warning of equipment failures;

4. Technology Service Providers: Package the method as SDKs or API interfaces to provide efficient log analysis solutions for small and medium-sized enterprises, driving the widespread adoption of log analysis technology.

VI. CONCLUSION

Addressing limitations in existing distributed log analysis—specifically inadequate feature extraction and weak capture of critical information within intelligent systems—this approach overcomes the difficulty of balancing processing accuracy and operational efficiency—this paper proposes a distributed log analysis method based on multi-scale convolutional attention mechanisms. It constructs a comprehensive

framework encompassing "log preprocessing—multi-scale feature extraction—dual-dimensional attention enhancement—lightweight inference analysis." Key research conclusions are as follows:

1. Established a preprocessing pipeline tailored to distributed log characteristics. Through structured transformation, noise filtering, and feature encoding, it lays the foundation for efficient analysis.

2. Designed a multi-scale convolutional feature extraction module. By utilizing convolutional kernels of varying sizes, it extracts local, short-range, and long-range features, achieving comprehensive coverage of multi-granularity log characteristics.

3. Proposed a dual-dimensional attention mechanism that enhances key feature weights from both field and sequence dimensions, improving the discernibility of sparse critical information;

4. Optimized lightweight inference and parallel computing workflows. Through feature dimensionality reduction, efficient classifier design, and GPU parallel optimization, a balance between accuracy and efficiency was achieved.

AUTHOR CONTRIBUTIONS

Conceptualization –Xinyue Liu, Debiao Luo, Weijie Wang, Ke Hu and Wen Yang; methodology – Xinyue Liu, Debiao Luo, Ke Hu and Wen Yang; investigation – Xinyue Liu, Debiao Luo, Weijie Wang, Ke Hu and Wen Yang; writing-original draft preparation – Xinyue Liu, Debiao Luo, Weijie Wang, Ke Hu and Wen Yang. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] O. Bamisile F, S. Kun, C. Ukwuoma C, et al., "Attention residual network with multi-scale convolution branch for efficient solar photovoltaic module defect classification," Solar Energy, vol. 307, pp. 114323-114323, 2026, doi: 10.1016/J.SOLENER.2026.114323.

[2] L. Chen, Y. He, X. Bai, et al., "Bearing fault diagnosis based on attention mechanism with multiscale convolution and gated recurrent units," Engineering Research Express, vol. 8, no. 1, pp. 015223-015223, 2026, doi: 10.1088/2631-8695/AE2E7F.

[3] X. Jiang, H. Luo, Y. Sun, et al., "Fast Anomaly Detection for IoT Services Based on Multisource Log Fusion," IEEE Internet of Things Journal, vol. 11, no. 6, pp. 9405-9419, 2024, doi: 10.1109/jiot.2023.3323620.

[4] Z. Hu, X. Zhang, and H. Xiong, "Two-stage attention network for fault diagnosis and retrieval of fault logs," Expert Systems with Applications, vol. 249, no. PartA, pp. 12, 2024, doi: 10.1016/j.eswa.2024.123365.

[5] G. Zhang and Y. Ru, "CRRE-YOLO: An Enhanced YOLOv11 Model with Efficient Local Attention and Multiscale Convolution for Rice Pest Detection," Applied Sciences, vol. 16, no. 1, pp. 352-352, 2025, doi: 10.3390/APP16010352.

[6] A. Vaswani, N. Shazeer, and N. Parmar, "Attention Is All You Need: Extensions with Multi-Scale Convolution for Distributed Log Sequence Analysis," Advances in Neural Information Processing Systems, vol. 36, pp. 5998-6008, 2023.

[7] R. Malekpour, T. Baghfalaki, M. Ganjali, et al., "Joint modeling of mixed skewed longitudinal responses using convolution of normal and log-normal distributions: a Bayesian approach," Communications in Statistics: Simulation & Computation, pp. 55(1), 2026, doi: 10.1080/03610918.2024.2401437.

[8] J. Lou, Q. Zhang, and Q. Lin, "Mining Invariants from Logs Using Multi-Scale Convolutional Attention for System Problem Detection," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 4, pp. 3890-3903, 2024.

[9] T. Chen, S. Kornblith, and M. Norouzi, "A Simple Framework for Contrastive Learning of Visual Representations: Adaptation with Multi-Scale Attention for Log Data," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 5, pp. 2890-2907, 2024.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate: Extension to Multi-Scale Log Sequence Modeling," Journal of Machine Learning Research, vol. 25, no. 102, pp. 1-32, 2024.

[11] Y. Li and B. Wu, "A Fuzzy MCDM-Based Deep Multi-View Clustering Approach for Large-Scale Multi-View Data Analysis," Symmetry, vol. 17, no. 8, pp. 1253, 2025, doi: 10.3390/sym17081253.

[12] V. Rathinapriya and J. Kalaivani, "Adaptive weighted feature fusion for multiscale atrous convolution-based 1DCNN with dilated LSTM-aided fake news detection using regional language text information.Expert Systems," 2024; 41(11), doi: 10.1111/exsy.13665.

[13] A. Chai, Z. Fang, M. Lian, et al., "Hi-MDTCN: Hierarchical Multi-Scale Dilated Temporal Convolutional Network for Tool Condition Monitoring," Sensors, vol. 25, no. 24, pp. 7603, 2025, doi: 10.3390/s25247603.

[14] M. Baisen, Q. Bo, L. Ali, et al., "Galaxy morphological classification using Dynamic Multiscale Attention Network," Monthly Notices of the Royal Astronomical Society, vol. (2), pp. 2, 2025, doi: 10.1093/mnras/staf1037.

[15] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping: Integration with Multi-Scale Attention for Log Feature Compression," Journal of Machine Learning Research, vol. 25, no. 89, pp. 1-28, 2024.