Transactions on Computational and Scientific Methods | Vol. 4, No. 5, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Transformer-Based Semantic Log Modeling for Robust and Explainable Cloud Security Analytics

Zhenyu Yang
University at Buffalo, Buffalo, USA
lomo.yeung@gmail.com

Abstract: This paper addresses the challenges of diverse data sources, complex structures, and hidden abnormal behaviors in cloud-based log environments by proposing a Transformer-based method for fine-grained log analysis and intrusion detection. The method builds a unified representation framework for multi-source logs and integrates a multi-dimensional context-aware mechanism to capture deep semantic relationships across different log types and enhance the modeling of cross-source behavioral paths. An attention-guided interpretation module is introduced to enable saliency-based localization and visual analysis of abnormal events, improving interpretability and controllability during detection. The modeling process combines source-type embeddings, positional encoding, and a global attention mechanism to establish a cross-time and cross-dimensional information fusion strategy, effectively representing long-term dependencies and key features in log sequences. In various sensitivity analysis experiments, the model shows strong stability and robustness under different learning rates, structural depths, sequence lengths, and anomaly ratios. Extensive experimental results demonstrate that the proposed method outperforms mainstream models across multiple metrics, achieving accurate detection while enabling automatic identification and focus on key events, thus providing strong support for efficient log modeling and intelligent security protection in cloud platforms.

Keywords: Transformer structure, log modeling, anomaly detection, attention mechanism

1. Introduction

With the widespread adoption of cloud computing technologies, more enterprises and institutions are migrating their critical services to cloud platforms. However, this resource-sharing and highly distributed environment also brings unprecedented security challenges [1], [2]. The attack surface in the cloud is much larger than in traditional computing models. Attack behaviors are often highly covert, rapidly propagated across layers, and exhibit complex logical paths. At the same time, modern cloud platforms generate massive volumes of system and application logs daily, which contain rich clues related to security events. Efficiently extracting potential threats from such redundant, heterogeneous, and multidimensional log data is a key challenge that urgently needs to be addressed in cloud security research [3].

Traditional intrusion detection methods rely heavily on feature engineering, rule matching, or statistical modeling. These methods often suffer from poor detection capabilities, limited adaptability, and high false alarm rates in the face of increasingly sophisticated attacks and the dynamic nature of cloud environments. On one hand, attackers continuously exploit unknown vulnerabilities and evasive variants to bypass existing rule-based systems. On the other hand, log data itself has complex structures, rich semantics, and strong contextual dependencies. Shallow models fail to capture the deep correlations between events. A more

expressive and flexible modeling approach is needed to mine hidden threat patterns from log sequences and enable fine-grained modeling and high-confidence detection of complex attack behaviors [4].

Recently, the Transformer architecture has demonstrated remarkable capabilities in sequential modeling within the field of natural language processing. It excels at capturing long-range dependencies and global contextual semantics. This mechanism is naturally suited for log data. Logs are typically sequential streams of events, and attack behaviors are often hidden within seemingly normal multi-hop paths. Introducing the Transformer mechanism allows for preserving the overall structure of logs while building more semantically linked representations between them. This enables a deeper understanding of complex attack chains and addresses the limitations of traditional models in local perception and long-term dependency modeling [5].

Furthermore, at the fine-grained detection level, the multi-head attention mechanism in Transformers offers inherent interpretability. It can identify key log segments or event features that the model focuses on during the analysis [6]. This not only improves detection accuracy but also provides meaningful references for follow-up security response. It supports effective tracing and rapid handling in cloud environments. Especially in complex cloud architectures with multi-tenancy and multi-layered resource sharing, fine-grained log perception and semantic modeling are essential. The Transformer mechanism can overcome the traditional bottlenecks in log analysis. It captures attack traces across dimensions, time, and layers, enhancing the detection of covert attacks and advanced persistent threats [7].

In summary, applying Transformer models to log analysis and intrusion detection in cloud platforms promotes the development of intelligent log understanding technologies. It also offers a new perspective for building robust cloud security defense systems. This approach represents a paradigm shift in log analysis and shows strong potential for constructing a universal security detection framework. It enables high adaptability and generalization in evolving cloud computing systems. This research direction aligns with trends in cloud-native architecture, intelligent operations, and automated security [8]. It has significant theoretical value and practical implications for enhancing cloud security, reducing false alarms and missed detections, and building trustworthy cloud services.

2. Related work

2.1 Transformer Architecture

The Transformer architecture was originally proposed for sequence-to-sequence tasks. Its core innovation lies in relying entirely on the attention mechanism to capture dependencies among elements in the input sequence. Unlike traditional recurrent or convolutional networks, the self-attention mechanism in Transformers processes all positions in the sequence in parallel [9]. This significantly improves training efficiency and enhances the ability to model long-range dependencies. The mechanism dynamically computes the attention weights between positions through the interaction of Query, Key, and Value vectors. This enables flexible and powerful feature representation [10]. The multi-head attention mechanism further strengthens the model's capacity to capture different semantic features from multiple subspaces. As a result, Transformers have achieved remarkable success in various natural language processing tasks [11].

Structurally, the Transformer consists of an encoder and a decoder. Each encoder layer contains a self-attention sublayer and a feed-forward neural network sublayer, both equipped with residual connections and layer normalization. This modular design facilitates the construction of deep models while ensuring stable information flow and controllable training [12]. In recent years, many Transformer variants have emerged. These include lightweight architectures, efficient attention mechanisms, and domain-specific structural adaptations. Transformers are now applied beyond language modeling to image processing, speech recognition, time series modeling, and other domains. In particular, Transformers show strong scalability and generalization when handling large-scale, high-dimensional input data [13].

In log analysis and security detection tasks, the global modeling capacity and context awareness of Transformers are especially critical. Log data is essentially a form of time series that contains redundant,

repetitive, and semantically ambiguous information. Traditional methods often struggle to capture behavior correlations over long time spans [14]. The global attention mechanism in Transformers allows the discovery of latent behavior paths and causal relationships between log events. Moreover, Transformers do not assume fixed structural patterns or linear dependencies in the data. This makes them more suitable for logs, which are complex, non-uniform, and semantically unstable. Applying Transformers to fine-grained log analysis fully leverages their sequence modeling strength and provides new perspectives for intelligent security detection in cloud environments [15].

2.2 Intrusion Detection Algorithms

Intrusion detection is a core component of network and system security. It has long attracted widespread attention from both academia and industry [16]. Traditional intrusion detection algorithms are mainly divided into signature-based methods and anomaly-based methods [17]. Signature-based methods rely on expert-defined rule sets or attack patterns for matching. They offer high efficiency and low false positives but fail to detect unknown or variant attacks [18], [19]. Anomaly-based methods build models of normal behavior to identify deviations. These methods have a certain level of generalization and can detect unknown threats [20]. However, they are often affected by data drift, feature selection, and model stability, leading to high false alarm rates. As attack types become increasingly diverse and covert in cloud environments, traditional static detection methods face limitations in adaptability, timeliness, and fine-grained modeling. More intelligent and dynamic detection mechanisms are urgently needed [21].

With the advancement of machine learning, many learning-based intrusion detection approaches have emerged. Supervised models such as decision trees, support vector machines, and naive Bayes classifiers learn the boundaries between normal and malicious behavior using labeled data. Unsupervised and semi-supervised methods, including clustering, autoencoders, and isolation forests, aim to discover hidden structures in data without relying on labels, enabling the detection of anomalous behaviors [22]. While effective in specific scenarios, most of these methods still rely on fixed features, static windows, or simplified data structures. This makes it difficult to capture the complex and dynamic characteristics of multi-source and multi-dimensional log data in cloud environments. Moreover, attack behaviors often occur in sequences or chains, which require models to capture temporal dependencies and contextual relationships. Traditional methods lack sufficient ability in this regard.

With the introduction of deep learning, intrusion detection has shifted from shallow feature extraction to deep semantic modeling. Convolutional neural networks and recurrent neural networks have been widely used for processing logs, network traffic, and system call sequences [23]. These approaches support end-to-end feature learning and behavior recognition. Recurrent neural networks, such as LSTM, attempt to model the evolution of attacks through temporal recurrence. However, challenges in efficiency and stability limit their application in long-sequence modeling. Recently, Transformer-based methods have gained attention. Their global attention and contextual modeling capabilities offer new strategies for reconstructing attack paths and identifying critical events. These methods eliminate the limitations of fixed windows and local perception. They enable more robust and interpretable intrusion detection models in complex log scenarios and promote the development of intelligent, explainable, and automated cloud security detection systems.

Recent studies on cloud-resource management and distributed operational intelligence further expand the system context for Transformer-based cloud security analytics. Constraint-aware resource matching for virtual machine placement [24], global context modeling and structure-guided queue-time prediction in large-scale cloud systems [25], and operational-state representation learning for distributed failure perception [26] show that cloud platforms require models that jointly capture resource constraints, runtime state transitions, and structural interactions across components. Reinforcement learning-based intelligent decision modeling for large-scale distributed systems [27] further connects detection results with adaptive operational decisions, which is important for transforming log-level anomaly signals into actionable security responses.

Graph-enhanced orchestration of enterprise ETL processes through collaborative reasoning between graph neural networks and large language models [28], multi-view behavioral modeling with trend regression for cloud resource prediction [29], structure-aware Transformer modeling with multi-scale fusion and boundary-guided prediction [30], and hierarchical communication-computation scheduling for cloud-edge federated learning [31] provide complementary methodological support for the present framework. These studies emphasize multi-view representation learning, structure-aware feature enhancement, boundary-sensitive prediction, and coordinated scheduling under heterogeneous cloud conditions. They support the use of multi-dimensional contextual awareness and attention-guided interpretation in this study, where log semantics, temporal dependencies, and cross-source operational cues are fused to improve intrusion detection robustness and explainability.

3. Method

This study proposes a Transformer-based fine-grained log analysis and intrusion detection framework designed to address the inherent concealment of attack behavior and the difficulty of log semantic modeling in cloud environments. This approach integrates two key innovations in its overall architecture: the Multi-Dimensional Contextual Awareness (MDCA) mechanism and the Attention-Guided Interpretability Module (AGIM). The MDCA effectively captures semantic dependencies between logs from different sources and hierarchies, enhancing the completeness of attack path modeling by introducing multi-source log joint modeling and position embedding enhancement. The AGIM leverages Transformer multi-head attention weights to focus on and enhance the visualization of key log events, improving the interpretability of the detection process and the controllability of security responses. This overall approach addresses the high-dimensional, heterogeneous, and dynamically evolving log data of cloud platforms, constructing a unified detection framework that balances modeling capabilities with practical applicability. The model architecture is shown in Figure 1.

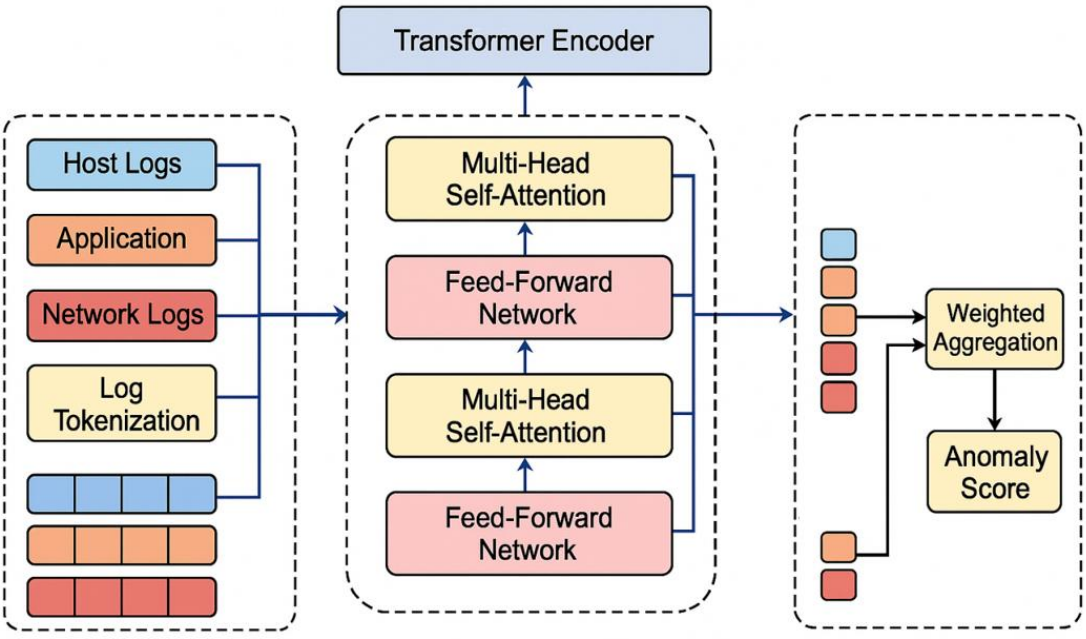


Figure 1. Overall model architecture diagram

3.1 Multi-Dimensional Contextual Awareness

This study introduces a multidimensional context-aware mechanism in the log modeling phase, aiming to fully capture the potential dependencies and semantic commonalities between different sources (such as host logs, application logs, and network logs). The model architecture is shown in Figure 2.

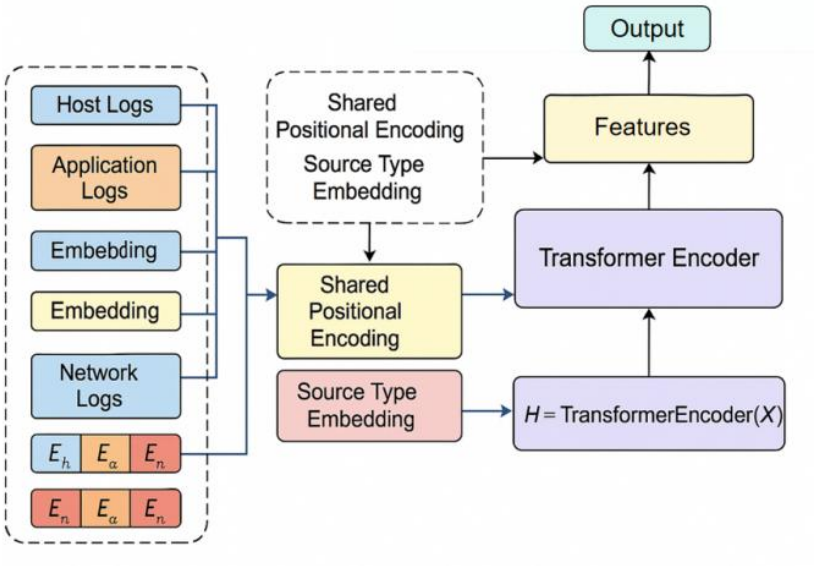


Figure 2. MDCA module architecture

Considering the high heterogeneity of log data in the cloud platform, we first embed and map various logs. Let the log sequences from different sources be $L^{(h)} = \{l_1^{(h)},...,l_n^{(h)}\}, L^{(a)} = \{l_1^{(a)},...,l_m^{(a)}\}, L^{(n)} = \{l_1^{(n)},...,l_k^{(n)}\}$, and the corresponding embedding representations are:

$$E^{(s)} = Embed(L^{(s)}), s \in \{h, a, n\}$$

To enhance cross-source modeling capabilities, we introduce shared position encoding and source feature encoding into the embedding results. The final representation of each log event is:

$$X_i = E_i + P_i + S_i$$

Where E_i is the embedding vector, P_i is the encoding based on timestamp or sequential position, and S_i is the source type embedding used to distinguish between host, application, and network dimensions.

We then jointly model sequences of multiple dimensions, construct a unified input sequence $X = [X^{(h)}, X^{(a)}, X^{(n)}]$, and input it into a multi-layer Transformer encoder to learn its global dependencies. The encoder output is recorded as:

$$H = Transforme\,rEncoder(X)$$

In this process, the self-attention mechanism models the degree of dependency of each pair of log events, which can be expressed as follows:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Where $Q = XW^Q, K = XW^K, V = XW^V$ represents the linear transformation of query, key, and value, respectively, and d_k represents the dimension of the key. Through this mechanism, the model can adaptively identify key interactions between logs from different sources.

To improve the ability to model cross-source event chains, we also introduce a source similarity adjustment term into the attention matrix and construct an enhanced attention calculation method as follows:

$$AugAttn(Q,K,V,S) = \text{softmax}(\frac{QK^T + \alpha \cdot sim(S_Q, S_K)}{\sqrt{d_k}})V$$

Where $sim(S_Q, S_K)$ represents the similarity between source embeddings, and α is the adjustment coefficient, which controls the influence of source context on attention distribution. This mechanism can guide the model to establish stronger dependencies between logs from the same or highly related sources, thereby improving the accuracy of modeling complex attack paths.

3.2 Attention-Guided Interpretability Module

To improve the interpretability of cloud environment log detection, this study designed an Attention-Guided Interpretability Module (AGIM) on top of the encoder. By aggregating multiple layers of multi-head attention, it generates event-level saliency distributions and fuses them with signals such as gradient attribution to produce stable and controllable interpretation results. The module architecture is shown in Figure 3.

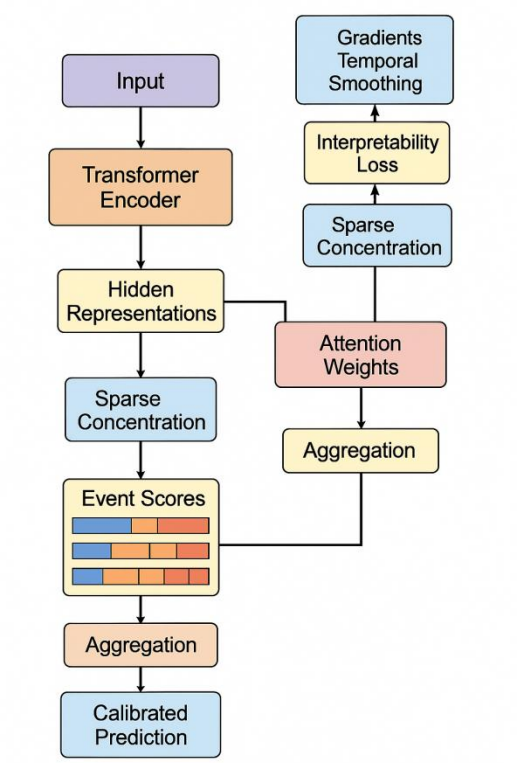


Figure 3. AGIM module architecture

First, let the attention of the h-th head in layer l be:

$$A^{(l,h)} = \text{softmax}(\frac{Q^{(l,h)}K^{(l,h)}}{\sqrt{d_k}})$$

At the log token level with sequence length N, the initial importance is obtained based on row-normalized attention accumulation:

$$s_i = \frac{1}{LH}\sum_{l=1}^{L}\sum_{h=1}^{H}\sum_{j=1}^{N} A_{ij}^{(l,h)}$$

And normalize it to get:

$$\hat{s}_i = \frac{\exp(s_i)}{\sum_{t=1}^{N} \exp(s_t)}$$

To avoid the bias caused by relying solely on attention, AGIM introduces a gradient attribution signal based on the target output and convexly combines it with the attention saliency. Let the scalar output of the model for the target category (or intrusion indicator) be $\hat{s}_i$, and the token-level gradient attribution be:

$$g_i = \| \frac{\partial \hat{y}}{\partial x_i} \|_2$$

And normalized to $g_i = \frac{g_i}{\sum_{t=1}^{N} g_t}$. The fusion importance of the two signals is defined as:

$$r_i = (1-\beta)s_i + \beta \tilde{g}_i, \beta \in [0,1]$$

Further gentle stretching of r_i to enhance head differentiation yields:

$$r_i = \frac{r_i^{\gamma}}{\sum_{t=1}^{N} r_t^{\gamma}}, \gamma > 0$$

To obtain sparse and continuous explanations, AGIM adds information entropy, sparse regularization, and temporal contiguity smoothing regularization to the training objectives. First, the entropy term that makes a small number of key events stand out:

$$L_{sparse} = \sum_{i=1}^{N} \tilde{r}_i \log \tilde{r}_i$$

Second, a time series smoothing term that encourages consistency in the interpretation of adjacent events:

$$L_{temp} = \sum_{i=1}^{N-1} (\tilde{r}_{i+1} - \tilde{r}_i)^2$$

After integrating task losses (such as detection or classification losses), the explanation-aware optimization objective of AGIM is:

$$L = L_{task} + \lambda_1 L_{sparse} + \lambda_2 L_{temp}$$

Where $\lambda_1, \lambda_2 > 0$ controls the trade-off between the explanation prior and the main task.

At the output level, AGIM combines token importance and hidden representation to form readable event-level and session-level explanation scores, and provides temperature-calibrated prediction distributions to improve confidence interpretability. Let the encoder's hidden representation be $\{h_i\}_{i=1}^{N}$, and first perform attention-weighted pooling to obtain:

$$z = W_{pool}(\sum_{i=1}^{N} r_i h_i) + b$$

Then use the temperature parameter $T > 0$ for calibration:

$$p = \text{softmax}(\frac{z}{T})$$

At the same time, the fragment/module-level explanation for operation and maintenance analysis can be given by the aggregation of the set $G \subseteq \{1,...,N\}$:

$$S_G = \frac{1}{|G|} \sum_{i \in G} \tilde{r}_i$$

Based on this, a multi-granularity explanation of the "Top-k key events" and "key link fragments" is returned to achieve traceable threat understanding and security decision support from point to chain.

4. Experimental Results

4.1 Dataset

This study uses the Los Alamos National Laboratory (LANL) Cyber Security Dataset as the primary data source. The dataset consists of logs collected over an extended period from a real enterprise-level internal network. It covers multi-dimensional security events such as hosts, users, processes, authentications, and network connections. It is one of the most widely used and publicly available high-quality datasets for network intrusion detection. The dataset provides accurate timestamps, unique identifiers, and entity relationships, making it suitable for various security tasks, including intrusion detection, anomaly behavior analysis, and log modeling.

The LANL dataset is highly sequential and structured. It contains event sequences collected from hundreds of machines and can be used to simulate behavior patterns within real attack chains. Its unified log format supports multi-source joint modeling. This makes it particularly suitable for evaluating the cross-dimensional modeling and context-aware mechanisms of Transformer-based approaches. During preprocessing, log events are parsed into multi-dimensional input sequences. These include user authentications, system calls, and network connections, which facilitate event-level modeling of attack paths.

The dataset also includes an annotated subset of attacks. This supports both supervised and weakly supervised intrusion detection research. It has a large volume and stable structure, and it realistically reflects the dynamics of complex network environments. These features allow models to be trained and evaluated in near-realistic conditions, improving their generalization ability and practical value.

4.2 Experimental setup

In the experimental setup, multi-source logs from the LANL dataset are first preprocessed and standardized. Host logs, network logs, and authentication logs are aligned by timestamp and parsed in a nested manner. Key fields are extracted as model inputs. All log events are organized into multi-dimensional event sequences in chronological order. Source type and positional encoding are embedded to preserve structural and contextual information. The training and testing sets are split using a sliding time window. This ensures causal consistency during sequence modeling and allows the model to predict events and detect anomalies without access to future information.

The model is trained based on a Transformer architecture. The input consists of joint log sequences, including embedded event vectors with position and source features. The Adam optimizer is used for training. Learning rate scheduling and weight regularization are applied to improve convergence stability. During training, attention weights and intermediate representations are recorded in real time and used for interpretability analysis. All experiments are conducted on a GPU-accelerated server to efficiently handle long sequences and high-dimensional attention mechanisms.

4.3 Experimental Results

1) Comparative experimental results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table 1: Comparative experimental results

Model	ACC	AUC	F1-score	Recall
ResNet [32]	87.41	89.26	85.33	83.57
ResNext [33]	88.02	90.14	86.05	84.90
ConvNext [34]	89.37	91.03	87.76	86.92
LSTM [35]	86.25	87.89	84.10	82.61
LSTM+Attention [36]	88.44	90.45	86.93	85.73
Ours	91.62	93.08	89.88	88.51

From the overall results, the proposed model outperforms all mainstream methods across all evaluation metrics. This demonstrates its significant advantages in log modeling and intrusion detection tasks. Compared with traditional convolution-based architectures such as ResNet, ResNeXt, and ConvNeXt, the proposed method captures contextual relationships and structural features across events more effectively. This highlights the global modeling capacity of Transformers in handling serialized log data. In terms of ACC and AUC, the proposed method achieves 91.62% and 93.08%, respectively. These represent over 2% improvement compared to ConvNeXt, indicating stronger overall classification accuracy and better confidence in detecting anomalous classes.

For time series models, LSTM and LSTM with attention offer certain strengths in modeling sequential features. However, they have limited ability to capture semantic associations across multi-source logs. Their performance is constrained by local windows and recursive structures, making it difficult to model complex long-range attack chains. In contrast, the proposed multi-dimensional context-aware mechanism and attention-guided explanation module enhance information interaction across log sources. This improves the model's ability to detect anomalous patterns and behavioral paths. Notably, the model achieves significant improvements in both Recall and F1-score.

The model achieves a Recall of 88.51%, showing its strong sensitivity and robustness in identifying anomalous events. This is especially important in intrusion detection, where missed detections often pose greater security risks than false alarms. The improved F1-score indicates a balanced performance between precision and recall. It avoids the instability seen in traditional methods that tend to favor one metric over the other. With the introduction of interpretability, the model not only detects anomalies but also pinpoints key log events through attention distributions. This enhances the interpretability and operational usability of the detection results.

In summary, the Transformer-driven architecture and attention-guided explanation module proposed in this work offer a more expressive representation for multi-source log input, fine-grained semantic modeling, and behavioral path reconstruction. This effectively addresses the detection blind spots of traditional models in complex cloud environments. It improves detection accuracy and provides valuable support for security auditing, threat tracing, and response optimization. The approach has strong potential for practical deployment.

2) Ablation Experiment Results

This paper also gives the ablation experiment results, which are shown in Table 2.

Table 2: Ablation Experiment Results

Method	ACC	AUC	F1-score	Recall
Baseline	88.37	89.90	86.04	84.10

+ MDCA	89.92	91.47	87.35	85.69
+ AGIM	90.31	91.86	87.90	86.23
+ All	91.62	93.08	89.88	88.51

As shown in Table 2, the two proposed core modules — Multi-Dimensional Context-Aware mechanism (MDCA) and Attention-Guided Interpretation Module (AGIM) — both bring notable performance improvements. This confirms their effectiveness in log modeling and intrusion detection tasks. Although the baseline model possesses basic Transformer encoding capabilities, it falls short in handling log heterogeneity and modeling behavior paths. As a result, it shows relatively low accuracy (ACC) and recall.

After integrating the MDCA module, the model's ACC increases to 89.92%, with improvements also observed in AUC and F1-score. This indicates that MDCA plays a key role in enhancing the modeling of relationships across different log sources. By embedding source types and positional information, MDCA improves the model's ability to perceive cross-dimensional behavior patterns. It is especially robust in handling complex attack chains and abnormal sequences. The observed gain in recall further demonstrates the module's strength in detecting potential anomalies and reducing missed detections.

Introducing the AGIM module alone also yields performance gains, particularly in F1-score and recall. This shows that guided attention not only strengthens the model's focus on critical log events but also enhances its ability to interpret behavior paths through saliency scoring and gradient-based fusion. The model becomes more sensitive and accurate in capturing anomalous signals. Moreover, this module lays a technical foundation for later interpretability analysis and security response.

When both modules are enabled together, the model achieves the best overall performance. This indicates that MDCA and AGIM are highly complementary. MDCA enhances semantic and structural modeling, while AGIM improves the detection and interpretation of key events. In complex cloud environments, log data often exhibits cross-source, sequential, and semantically sparse characteristics. This combined architecture significantly improves model performance in fine-grained log analysis and intrusion detection without sacrificing computational efficiency. It demonstrates both the rationality and practicality of the design.

3) Analysis of the impact of different learning rate settings on model performance

This paper also analyzes the impact of different learning rate settings on model performance, and the experimental results are shown in Figure 4.

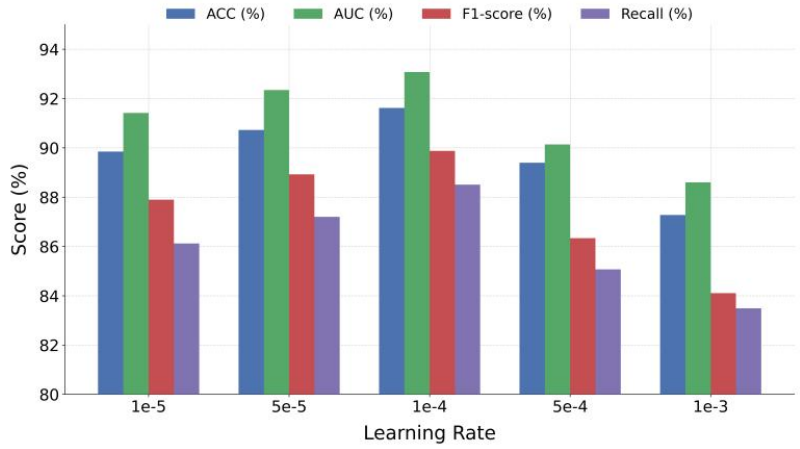


Figure 4. Analysis of the impact of different learning rate settings on model performance

As shown in Figure 4, the learning rate has a significant impact on model performance, especially in terms of ACC and AUC. The model achieves optimal performance when the learning rate is set between 1×10^{-4} and

5×10^{-5} . This indicates that within this range, the model strikes a good balance between convergence speed and training stability. A lower learning rate, such as 1×10^{-5} , ensures stable training but limits performance improvement. In contrast, a higher learning rate, such as 1×10^{-3}, leads to unstable training and a notable drop in final metrics.

It is worth noting that the AUC curve shows more fluctuation across the learning rate range. This suggests that the decision boundary between positive and negative samples is highly sensitive to the learning rate. Specifically, AUC reaches its peak between 5×10^{-5} and 1×10^{-4} . This indicates stronger anomaly detection capability at this stage. The result aligns with the characteristics of log sequences, where attack patterns are often sparse and irregular. The trends of F1-score and Recall follow a similar pattern, reflecting the direct influence of learning rate on the model's ability to capture anomalous events.

With the integration of the Multi-Dimensional Context-Aware mechanism (MDCA) and the Attention-Guided Interpretation Module (AGIM), model performance relies on capturing long-range dependencies and focusing on key events. These capabilities require effective training, which in turn depends heavily on a well-tuned learning rate. Without this, critical components such as multi-head attention and positional encoding cannot be properly optimized. This would degrade the model's overall representation capacity. The experimental results validate the necessity of fine-tuning optimization parameters in complex log analysis tasks.

The final results show that in fine-grained log intrusion detection, the learning rate acts as a key hyperparameter. It not only influences basic accuracy but also affects the model's ability to learn semantic associations across sources. Selecting an appropriate learning rate within a reasonable range enhances detection performance. It also ensures stable convergence and training robustness, thereby supporting deployability in large-scale cloud environments.

4) The impact of changes in the number of Transformer layers on detection results

This paper also gives the impact of changing the number of Transformer layers on the detection effect, and the experimental results are shown in Figure 5.

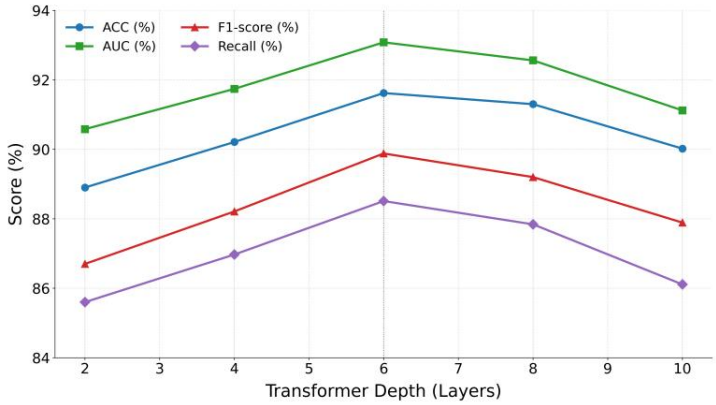


Figure 5. The impact of changes in the number of Transformer layers on detection results

As shown in Figure 5, the number of Transformer layers has a significant impact on detection performance. When the number of layers is low (between 2 and 4), the model lacks sufficient contextual modeling capability. ACC and AUC scores remain relatively low, indicating that the model is not yet effective in capturing complex behavioral chains across multi-source logs. As the number of layers increases to 6, the model reaches peak performance across all evaluation metrics. This suggests that a good balance is achieved between structural depth and semantic extraction, enabling the model to capture both potential attack behaviors and contextual features more comprehensively.

When the number of layers increases further to 8 and 10, model performance begins to decline slightly. This trend indicates that deeper structures may introduce redundant modeling and overfitting risks in log-based scenarios. The decline is more noticeable in F1-score and Recall, suggesting that the model's ability to detect anomalies does not grow linearly with increased depth. This may be due to the sparsity and noise often present in the sequential structure of log data in cloud environments. Without a large amount of high-quality labeled samples, deeper Transformer structures may reduce the model's focus on key patterns.

With the integration of the Multi-Dimensional Context-Aware mechanism (MDCA) and the Attention-Guided Interpretation Module (AGIM), moderate network depth becomes critical for capturing long-term dependencies and cross-dimensional interactions in heterogeneous logs. The model achieves optimal performance with six Transformer layers. This depth supports global representation of log events while maintaining high representation efficiency and gradient stability. It contributes to the unified goal of anomaly detection and interpretability.

These results further validate that structural design in fine-grained log modeling should be guided by task-specific characteristics rather than blindly increasing network depth. In intrusion detection tasks characterized by semantic sparsity and behavioral discreteness, a controllable depth and interpretable mechanisms are more conducive to sustained performance gains. This emphasizes the practical value and methodological innovation of this study in architectural tuning and mechanism integration.

5) The impact of log sequence length on detection accuracy

This paper also explores the impact of log sequence length on detection accuracy, considering it as a key factor influencing the model's ability to capture contextual information and behavioral patterns within multi-source log data. Since log sequences often contain time-dependent and semantically correlated events, the length of the input sequence directly affects how much temporal and contextual information the model can utilize during training and inference. By varying the sequence length, the study systematically examines how the model's representation capacity and detection performance respond to changes in the amount of available historical information. This analysis provides important insights into the trade-off between contextual richness and computational complexity, and highlights the necessity of selecting an appropriate sequence length to ensure stable and efficient detection under real-world cloud environments. The detailed experimental settings and observations are illustrated in Figure 6.

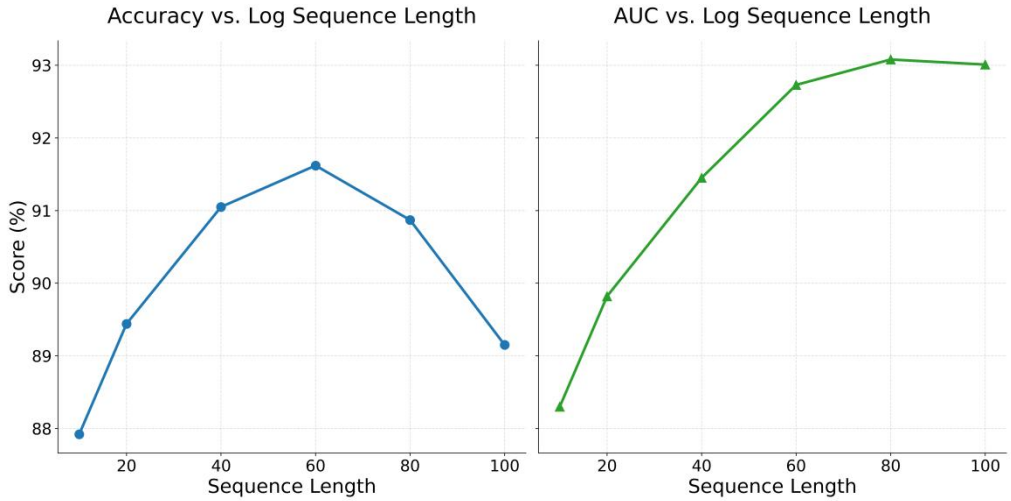


Figure 6. The impact of log sequence length on detection accuracy

As shown in Figure 6, the length of the log sequence has a clear impact on detection performance. This is especially evident in the trends of accuracy and AUC. As the sequence length increases from 10 to 60, the

overall performance steadily improves. This indicates that in multi-source log scenarios, increasing the sequence length provides richer contextual information. It helps the model learn potential behavior patterns and semantic associations within attack chains, improving its ability to identify anomalous events.

Accuracy (ACC) reaches its peak when the sequence length is 60, then slightly declines. This trend suggests that excessively long sequences may introduce redundant or irrelevant information. This can dilute the model's attention and reduce its ability to focus on critical events. In cloud environments, logs often contain a large number of background actions and normal events. Processing longer sequences may increase the model's burden and raise the risk of false detections. In contrast, moderate sequence lengths help the model capture meaningful behavioral context and yield better structural representations.

In comparison, the AUC curve shows a consistent upward trend across the full sequence length range. It gradually saturates when the length approaches 80. This shows that as the model receives more context, its ability to distinguish between positive and negative samples improves. The robustness of the model increases, especially when dealing with imbalanced data or unclear attack behaviors. This trend suggests that AUC is more sensitive to global representation capabilities and less affected by short-term fluctuations or local noise.

With the integration of the Multi-Dimensional Context-Aware mechanism (MDCA) and the Attention-Guided Interpretation Module (AGIM), the results validate the fundamental role of sequence modeling in anomaly detection. A moderate sequence length ensures sufficient contextual coverage while preventing over-dilution of attention. It also supports high-quality representations for the interpretability modules. These findings highlight the importance of log preprocessing and sequence truncation strategies in model training. They also reflect the flexibility and adaptability of the proposed method in real-world deployment scenarios.

6) Analysis of the impact of abnormal proportion changes on model robustness

Finally, this paper presents an analysis of the impact of abnormal ratio changes on model robustness, and the experimental results are shown in Figure 7.

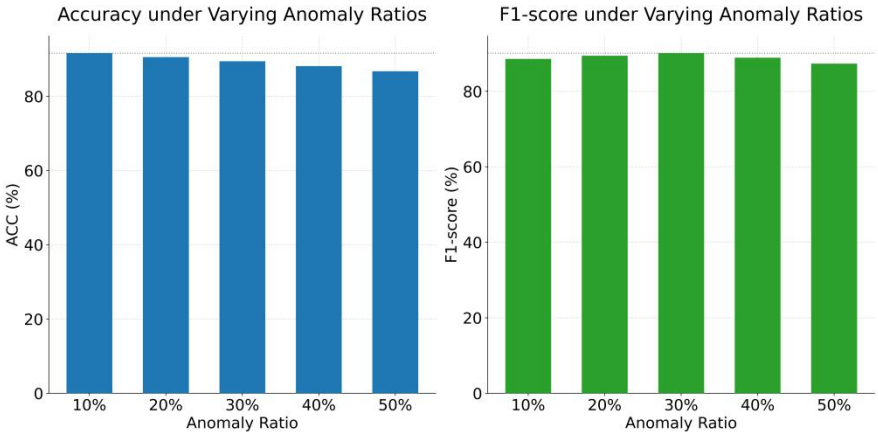


Figure 7. Analysis of the impact of abnormal proportion changes on model robustness

As shown in Figure 7, the anomaly ratio has a clear impact on model robustness. This is especially evident in the distinct fluctuation trends of accuracy (ACC) and F1-score. As the anomaly ratio increases from 10 percent to 50 percent, ACC shows a general downward trend. This indicates that the overall classification accuracy is affected when the model encounters a high proportion of anomalous data. This may be due to the semantic diversity or unstructured nature of anomaly samples, which can shift the model's attention and reduce its precision in identifying normal events.

The trend in F1-score is more consistent. It reaches a peak when the anomaly ratio is between 20 percent and 30 percent, then gradually declines. This suggests that a moderate level of anomalies helps the model learn abnormal patterns more effectively. It also achieves a better balance between precision and recall. However, a

high proportion of anomalies may lead to overfitting on abnormal features. This reduces the model's generalization ability for normal behaviors and lowers its overall detection performance. The results also confirm the model's sensitivity to imbalanced data distributions.

It is worth noting that the proposed Multi-Dimensional Context-Aware mechanism (MDCA) and Attention-Guided Interpretation Module (AGIM) enhance the model's structural awareness and its ability to focus on local events. These mechanisms allow the model to maintain a relatively high F1-score even as the anomaly ratio increases. This shows that the model is robust and capable of adapting to dynamic changes in data distribution, meeting the fault-tolerance requirements of real-world deployments.

In summary, the experimental results highlight the importance of controlling the anomaly sample ratio during training. A well-chosen ratio not only improves model performance but also strengthens its adaptability to abnormal surges or data skew in practical cloud environments. The findings also confirm that the proposed method performs well not only in static evaluation metrics but also in terms of robustness under dynamic conditions. This demonstrates the structural adaptability of the model.

5. Conclusion

This paper proposes a Transformer-based method for fine-grained log analysis and intrusion detection. It is designed to address the complex requirements of modeling multi-source heterogeneous logs and identifying abnormal behaviors in cloud environments. The method integrates a multi-dimensional context-aware mechanism and an attention-guided interpretation module. It captures global behavioral paths while enabling precise identification and visual interpretation of key log events. Through systematic architectural design and information fusion strategies, the model overcomes the limitations of traditional methods in sequence modeling depth and behavior interpretability. It demonstrates strong representational capacity and practical usability.

Across multiple experimental dimensions, the method achieves superior performance in accuracy, AUC, and F1-score. This confirms its effectiveness and robustness in intrusion detection tasks. It maintains stable recognition performance under different settings, including variations in sequence length, model depth, and anomaly ratio. The attention-guided module not only improves detection accuracy but also provides transparent and interpretable support for security operations. It helps with auditing and anomaly tracing.

This research contributes to the integration of log modeling and security detection from a technical perspective. It also offers practical value for intelligent operations, security monitoring, and attack tracing in real-world cloud platforms. The method can be applied to large-scale log stream processing in multi-tenant environments. It is also adaptable to other complex security analysis tasks, such as industrial control system logs and edge device event monitoring. It shows strong cross-platform adaptability and engineering potential.

Future work may explore the model's capacity for anomaly detection in unsupervised or few-shot scenarios. Techniques such as generative adversarial learning and self-supervised pretraining can be introduced to enhance generalization and transferability. In addition, the method may be combined with federated learning and edge intelligence to support distributed and privacy-sensitive log analysis tasks. Improving computational efficiency and deployment flexibility in high-throughput environments will be key to enabling large-scale applications.

References

[1] H. Guo, S. Yuan, and X. Wu, "LogBERT: Log anomaly detection via BERT," in Proc. 2021 Int. Joint Conf. Neural Networks (IJCNN), 2021, pp. 1-8.

[2] H. Guo, X. Lin, J. Yang, et al., "TransLog: A unified transformer-based framework for log anomaly detection," arXiv preprint arXiv:2201.00016, 2021.

[3] H. Guo, J. Yang, J. Liu, J. Bai, B. Wang, Z. Li, T. Zheng, B. Zhang, J. Peng and Q. Tian, "LogFormer: A Pre-train and Tuning Pipeline for Log Anomaly Detection," arXiv preprint, arXiv:2401.04749, 2024.

[4] H. He, X. Li, P. Chen, et al., "Efficiently localizing system anomalies for cloud infrastructures: A novel dynamic graph transformer based parallel framework," Journal of Cloud Computing, vol. 13, no. 1, p. 115, 2024.

[5] Z. Long, H. Yan, G. Shen, et al., "A Transformer-based network intrusion detection approach for cloud security," Journal of Cloud Computing, vol. 13, no. 1, p. 5, 2024.

[6] Y. Xie, H. Zhang, and M. A. Babar, "LogGD: Detecting anomalies from system logs with graph neural networks," in Proc. 2022 IEEE 22nd Int. Conf. Software Quality, Reliability and Security (QRS), 2022, pp. 299-310.

[7] X. Wei, J. Wang, C. Sun et al., "Log-based Anomaly Detection for Distributed Systems: State of the Art, Industry Experience, and Open Issues," Journal of Software: Evolution and Process, 2024.

[8] Y. Liu, S. Ren, X. Wang et al., "Temporal Logical Attention Network for Log-Based Anomaly Detection in Distributed Systems," Sensors, 2024.

[9] H. Guo, J. Yang, J. Liu et al., "LogFormer: A Pre-train and Tuning Pipeline for Log Anomaly Detection," Proceedings of the AAAI Conference on Artificial Intelligence, AAAI Publications, 2024.

[10] X. Wu, H. Li and F. Khomh, "What Information Contributes to Log-based Anomaly Detection? Insights from a Configurable Transformer-Based Approach," arXiv preprint, arXiv:2409.20503, 2024.

[11] M. Ma, L. Han and C. Zhou, "Research and Application of Transformer Based Anomaly Detection Model: A Literature Review," arXiv preprint, 2024.

[12] A. Shahwali, "Using generative pre-trained transformers to detect anomalous telecommunication logs," 2024.

[13] K. Gong, S. Luo, L. Pan, L. Zhang, Y. Zhang and H. Yu, "LogETA: Time-aware Cross-system Log-based Anomaly Detection with Inter-class Boundary Optimization," Future Generation Computer Systems, vol. 157, pp. 16–28, 2024.

[14] M. Alshomrani, A. Albeshri, B. Alturki, et al., "Survey of transformer-based malicious software detection systems," Electronics, vol. 13, no. 23, p. 4677, 2024.

[15] G. De La Torre Parra, L. Selvera, J. Khoury, et al., "Interpretable federated transformer log learning for cloud threat forensics," in Proc. Network and Distributed System Security Symp. (NDSS), 2022.

[16] H. Kheddar et al., "Transformers and Large Language Models for Efficient Intrusion Detection Systems: A Survey," arXiv preprint, 2024.

[17] W. Sakong, J. Kwon, K. Min, et al., "Anomaly transformer ensemble model for cloud data anomaly detection," IEEE Transactions on Cloud Computing, 2024.

[18] Y. Liu, S. Ren, X. Wang, et al., "Temporal logical attention network for log-based anomaly detection in distributed systems," Sensors, vol. 24, no. 24, p. 7949, 2024.

[19] J. Doshi, "Live log analysis using integrated SIEM and IDS using machine learning," dissertation, Institute of Technology, 2024.

[20] H. Thompson, S. J. Stolfo, A. D. Keromytis, et al., "Anomaly Detection at Multiple Scales (ADAMS)," Tech. Rep., 2011.

[21] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19-31, 2016.

[22] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms: The Numenta anomaly benchmark," in Proc. 2015 IEEE 14th Int. Conf. Machine Learning and Applications (ICMLA), 2015, pp. 38-44.

[23] T. Xiao, Z. Quan, Z. J. Wang, et al., "Loader: A log anomaly detector based on transformer," IEEE Transactions on Services Computing, vol. 16, no. 5, pp. 3479-3492, 2023.

[24] J. Kou and E. Pei, "Constraint-aware resource matching for virtual machine placement in cloud environments," 2024.

[25] H. Zhuang and K. Gao, "Global context modeling and structure-guided feature enhancement for queue time prediction in large-scale cloud systems," 2024.

[26] W. Huang, R. Zhan, and Z. Huang, "Data-driven failure perception for distributed systems through operational state representation learning," 2024.

[27] R. Wei and L. Tan, "Reinforcement learning-based intelligent decision modeling for large-scale distributed systems," 2024.

[28] Y. Zhang and L. Steven, "Automatic orchestration of enterprise ETL processes via graph neural network and large language model collaborative reasoning," 2024.

[29] X. Zhang and Z. Fang, "Deep learning-based multi-view behavioral modeling and trend regression for cloud resource prediction," 2024.

[30] K. Zeng, "Transformer-based structure-aware lung cancer image segmentation with multi-scale fusion and boundary-guided prediction," 2024.

[31] R. Meng, "Hierarchical communication and computation scheduling for efficient federated learning in cloud-edge collaborative intelligent systems," 2024.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[33] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1492-1500.

[34] Z. Liu, H. Mao, C. Y. Wu, et al., "A ConvNet for the 2020s," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11976-11986.

[35] J. Brownlee, Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models With Deep Learning. Machine Learning Mastery, 2017.

[36] B. Al-Omar and Z. Trabelsi, "Intrusion detection using attention-based CNN-LSTM model," in Proc. IFIP Int. Conf. Artificial Intelligence Applications and Innovations, Cham, Switzerland: Springer Nature Switzerland, 2023, pp. 515-526.