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

Temporal-Semantic Graph Attention Networks for Cloud Anomaly Recognition

Heyi Wang
Illinois Institute of Technology, Chicago, USA
helenwangheyi@gmail.com

Abstract: This paper proposes an intelligent modeling framework for classifying high-dimensional, dynamic, and heterogeneous memory access behaviors in cloud computing environments. The method takes memory access sequences as input and applies a structure-enhanced attention mechanism to extract hidden dependencies. This improves the model's ability to capture key semantics within the temporal information. At the same time, a dynamic semantic knowledge graph is constructed to link access events with contextual entities such as tenants, services, and tasks. A temporal dimension is also introduced to build a graph representation that updates in real-time as the system evolves. In the model design, structural attention weights enable context-aware behavior classification. The dynamic integration of the semantic graph further enhances the model's ability to understand and classify complex behavior paths. Based on this architecture, the paper conducts comparative experiments, ablation analysis, and hyperparameter sensitivity evaluations using several public datasets. The results confirm the advantages of the proposed method in terms of F1-Score, AUC-ROC, and Accuracy. Experimental findings show that the model effectively detects implicit anomalies in multi-tenant systems. It also reduces false positive rates and improves the recognition of complex behavior patterns. The model demonstrates strong stability and robustness. The framework is end-to-end and is suitable for high-dimensional time-series classification tasks in dynamic settings. It enhances the intelligence level of behavior recognition and risk control in cloud platforms.

Keywords: Structural attention mechanism; semantic graph modeling; memory access behavior; high-dimensional temporal discrimination

1. Introduction

The rapid development of cloud computing infrastructure has driven the centralized deployment of critical services and computing tasks in virtualized environments[1]. This trend raises higher demands for precise scheduling of computing resources and deep security monitoring. In such environments, memory plays a key role as a shared computing resource among multiple tenants. Memory access behavior carries highly concentrated information. Especially under Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) architectures, memory access patterns in the cloud reveal tenant behaviors, task execution paths, and potential system risks. Therefore, analyzing access behavior at the memory level has become essential for ensuring resource security and platform stability. However, due to the complexity of task scheduling and resource sharing in modern cloud platforms, traditional static rule-based detection or low-dimensional statistical methods fail to capture subtle distinctions between normal and abnormal memory behaviors[2].

In cloud security, memory-level threats are highly concealed and dynamic. Attackers may construct sophisticated memory access patterns such as side-channel attacks, cache poisoning, or row hammers to

evade traditional intrusion detection systems. At the same time, some non-malicious but anomalous behaviors like burst load or rescheduling can also lead to changes in memory access characteristics. These may cause false positives or negatives in detection systems. Thus, building a discriminative mechanism with contextual awareness and semantic understanding in high-dimensional, strongly coupled, and heterogeneous memory access time series has become a major challenge. This requires models with a deep understanding of sequential patterns and the ability to dynamically link access behaviors with system states and tenant contexts[3,4].

In this context, knowledge graph techniques have shown unique advantages in expressing complex entity relationships and semantic constraints. They have become important tools for modeling heterogeneous relationship networks in cloud systems[5]. By representing memory access behaviors together with tenants, services, nodes, tasks, and time windows in a unified graph structure, one can systematically reveal the structural and semantic associations behind access activities. Meanwhile, the development of attention mechanisms in deep learning enables the effective capture of keyframes, local patterns, and long-range dependencies in high-dimensional time series. This significantly enhances discriminative performance in complex behavior sequence anomaly detection. Therefore, integrating knowledge graphs with attention mechanisms offers a new approach to addressing memory access anomaly detection in cloud environments[6].

Furthermore, the diversity of services, dynamic scheduling, and tenant heterogeneity in cloud platforms make it impossible to rely on single semantic labels or discrete event definitions. Many potential threats may not show obvious statistical deviations at the early stage. However, their behavioral paths may already disturb system structures. For instance, some attacks build long access chains and mimic normal behavior patterns[7,8]. These may leave traces of abnormal access long before performance indicators change. Modeling access behavior based on contextual structure and introducing multi-granularity attention mechanisms along the time axis can help identify such structurally hidden yet behaviorally abnormal threat paths at an earlier stage. This is critical for building a forward-looking and robust cloud security defense system.

In summary, for anomaly detection in memory access patterns within cloud environments, it is essential to build an intelligent model that integrates high-dimensional time-series modeling with complex structural semantic understanding. This enhances the precision in detecting malicious behavior. It also improves the platform's ability to sense risks such as abnormal resource usage, policy misconfiguration, and system degradation. As cloud infrastructure evolves and system complexity increases, exploring AI models with knowledge enhancement and semantic adaptability will be key to advancing the security, stability, and intelligence of cloud platforms.

2. Related work

2.1 Attention Mechanism

The attention mechanism was initially proposed to enhance sequence modeling capabilities. It aims to address the problems of gradient vanishing and information loss in traditional neural networks when processing long sequences. In temporal modeling tasks, attention assigns different weights to various positions in the input sequence[9,10]. This allows the model to automatically focus on the most relevant information for the current task. As a result, it significantly improves the modeling of complex dependencies. Attention is especially effective for data with strong temporal dependencies and high-dimensional dynamic features. It enhances the model's ability to represent the global context. This has led to widespread application in fields such as natural language processing and speech recognition. In recent years, attention mechanisms have evolved into various forms, including self-attention, cross-attention, and multi-head attention. These developments have expanded its applicability to more diverse scenarios and data structures[11].

In cloud computing environments, attention mechanisms demonstrate strong adaptability and expressive power. They play a key role in tasks such as resource monitoring, behavior modeling, and anomaly

detection[12]. Cloud platforms are highly dynamic. Resource scheduling is frequent. The interactions and behavior sequences between services often show complex nonlinear evolution. Traditional statistical methods or fixed-window sequence models often fail to capture subtle changes and contextual dependencies in behavior patterns. With attention mechanisms, models can selectively focus on the most relevant parts of historical sequences according to the current decision task. This improves both the accuracy and interpretability of anomaly detection. Furthermore, when combined with multi-scale modeling and hierarchical attention, the model can extract multi-granularity features from behavior sequences. This provides richer support for high-dimensional time-series classification tasks[13].

As understanding of the attention mechanism deepens, its capacity to handle both structured and unstructured data continues to grow. In graph-based applications, attention can capture structural relationships between nodes[14,15]. It also allows the model to evaluate edge weights and contextual importance. This enables fine-grained modeling of complex interaction structures. In the field of cloud security, especially in modeling memory-level access behaviors, access sequences often carry complex semantics and contextual dependencies. These include tenant identities, scheduling nodes, and task properties[16]. Attention mechanisms help establish effective associations among high-dimensional and heterogeneous features. This enhances sensitivity and robustness in abnormal pattern recognition. Such capabilities are particularly critical for detecting threats that rely on fine-grained timing and behavioral patterns, such as side-channel attacks or cache interference.

In addition, the integration of attention mechanisms with other deep learning modules continues to advance. This has driven the development of cross-modal and multi-source joint modeling approaches. For example, in modeling multi-modal system observability, attention helps represent complementary relationships among metrics, logs, and topology data. In tasks involving semantic enhancement through knowledge graphs, attention allows the model to dynamically focus on the most relevant graph structures or entity relations. This improves the semantic interpretability of abnormal behavior. The flexibility of such combinations makes attention not only a tool for sequence modeling but also a general framework for understanding complex system behaviors. Applying it to anomaly detection of memory access patterns in cloud platforms enhances not only the methodological level but also the semantic and decision-making capabilities of the entire detection system.

2.2 Knowledge Graph

As a structured semantic network that represents entities and their relationships, knowledge graphs have played an increasingly important role in intelligent systems in recent years[17]. The core idea is to connect semantically clear entity nodes with relational edges using a graph structure. This builds a knowledge representation framework that is interpretable, inferable, and scalable[18]. Compared to traditional vector space modeling methods, knowledge graphs provide a more natural way to represent heterogeneous information, multi-source data, and contextual dependencies in complex systems. They are particularly suitable for describing large-scale systems composed of multi-layer structures, various entity types, and dynamic interactions. In cloud computing environments, this modeling approach is especially effective for capturing the dynamic coupling among tenants, services, and resources. It helps integrate scattered information into a unified semantic representation space[19].

In cloud platform security and operations scenarios, the relationships among entities are highly complex. These include, but are not limited to, interactions between tenants and virtual machines, tasks and containers, services and nodes, and events and metrics. Traditional log analysis or metric monitoring methods struggle to establish systematic connections in such heterogeneous and semantically ambiguous data. With the introduction of knowledge graphs, this data can be mapped into a graph structure with explicit semantic relationships. This expands the understanding of system behavior from isolated points to broader surfaces[20]. It also supports the construction of event chains, causal chains, and even potential threat chains. These semantic structures provide crucial support for downstream classification and detection tasks. For example, by associating memory access behaviors with service structures, tenant contexts, and security policies, a

complete behavioral context can be constructed. This significantly improves system perception in the presence of vague, sparse, or novel threats.

Moreover, knowledge graphs possess inherent scalability and reasoning capabilities. They support the discovery of hidden associations and the completion of missing information based on existing knowledge. This is particularly critical in real-world cloud security applications. When facing new attack patterns or unseen behavior types, knowledge graphs can estimate unknown relationships through graph propagation and reasoning. This helps models maintain strong detection performance even in incomplete or noisy data environments. Especially in high-dimensional time-series scenarios, behavioral evolution often involves the joint dynamics of multiple entities and events. Knowledge graphs offer a stable semantic support framework. This helps models maintain structural consistency and semantic coherence when handling long sequences and multi-hop dependencies. These features are essential for detecting complex attack paths and identifying chained abnormal behaviors[21,22].

In recent years, the integration of graph neural networks and attention mechanisms has further enhanced the modeling power of knowledge graphs. Graph neural networks enable high-dimensional embedding of graph structures. Attention mechanisms allow dynamic weighting of nodes and relationships. Together, they support more flexible and precise graph semantic modeling. This approach improves the expressive power of knowledge graphs. It also allows joint modeling with other modalities such as time-series data, textual information, and metric sequences. In the context of anomaly detection in memory access patterns on cloud platforms, knowledge graphs help the model understand not only what access occurred, but also why it occurred and which system entities were involved. This leads to more comprehensive semantic-level decisions. This structured and semantic-aware modeling approach brings new perspectives and directions for cloud security research.

3. Method

This study proposes a high-dimensional temporal discrimination model that integrates the attention mechanism and knowledge graph to identify abnormal patterns of memory layer access behavior in cloud computing environments. This method deeply models access behavior from two dimensions: structure and timing, and has two key innovations. First, the Structure-Enhanced Attention Mechanism (SEAM) is introduced to improve the model's perception of fine-grained differences in behavior by jointly modeling access sequences and their contextual dependencies; second, a Dynamic Semantic Knowledge Graph (DSKG) is constructed to semantically connect memory access behavior with multiple entities such as tenants, services, tasks, and time, so that the model can structuredly express and reason about complex access behavior paths. This dual innovation provides a new modeling paradigm and semantic support for the intelligent discrimination of abnormal access patterns in multi-tenant, high-dimensional heterogeneous scenarios on cloud platforms. The detailed structure of the proposed model is illustrated in Figure 1.

3.1 Structure-Enhanced Attention Mechanism

Structure-Enhanced Attention Mechanism (SEAM) aims to improve the model's structural perception ability when processing high-dimensional time series data. It is particularly suitable for discrimination tasks with complex contextual dependencies and semantic sparsity problems in memory access sequences. Although the traditional self-attention mechanism performs well in capturing long-distance dependencies between elements in a sequence, it often ignores the structural information behind the sequence, such as resource allocation in a multi-tenant environment, topological relationships between services, and contextual constraints on access paths. To this end, SEAM introduces a structural modulation factor based on the standard attention mechanism to dynamically integrate the semantic structure and entity relationship on which the access behavior depends in the attention calculation. Its goal is to align the original behavior

representation to the structural semantic space to enhance the sensitivity and generalization ability of the discrimination task. Its module architecture is shown in Figure 2.

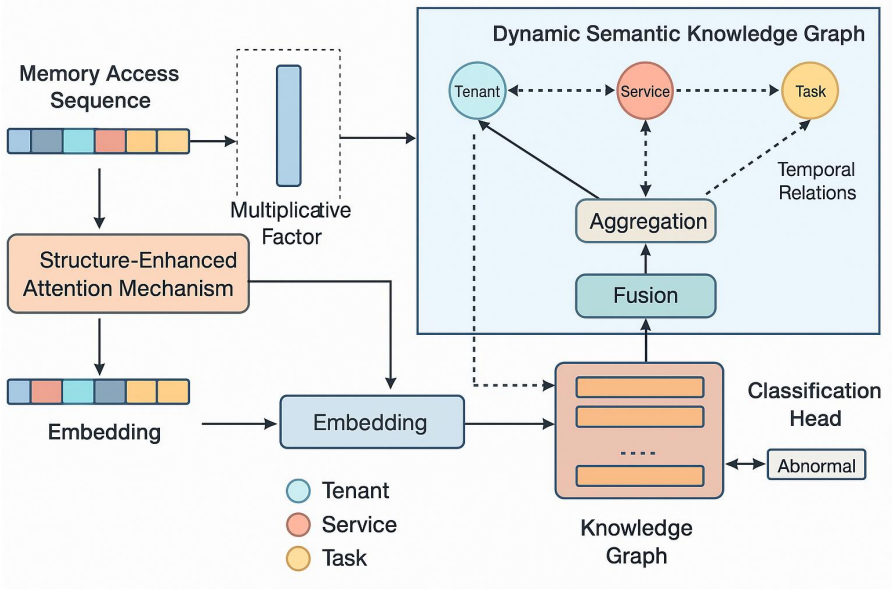


Figure 1. Overall model architecture diagram

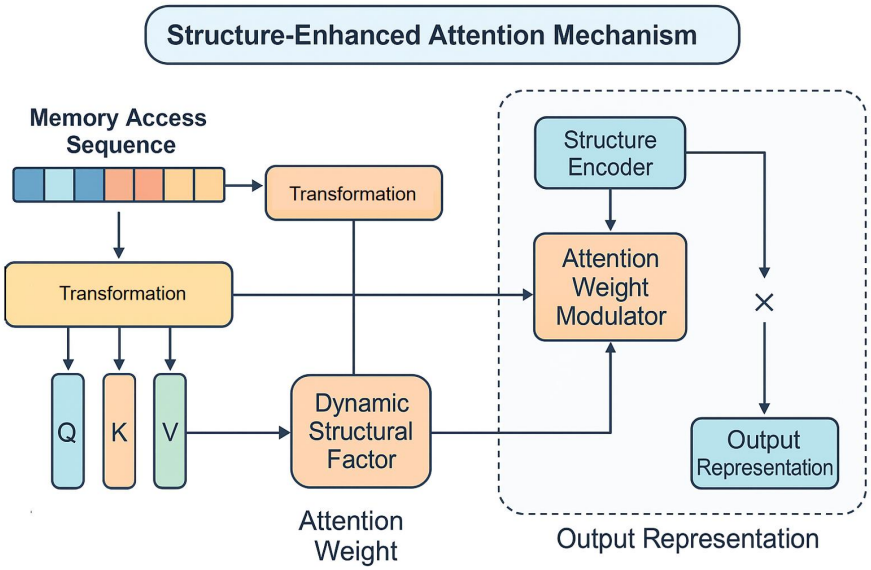


Figure 2. SEAM module architecture

Specifically, given a memory access sequence representation $X = [x_1, x_2, ..., x_n]$, where each x_i represents the feature vector of an access event, SEAM first generates query, key, and value matrices through linear transformation:

$$Q = XW_Q, K = XW_K, V = XW_V$$

Where $W_Q, W_K, W_V \in R^{d \times d'}$ is a trainable parameter. Then calculate the basic attention weight:

$$a_{ij} = \frac{\exp(Q_i \cdot K_j^T)}{\sum_{k=1}^{n} \exp(Q_i \cdot K_k^T)}$$

This weight is modeled only based on content similarity and fails to reflect the structural dependencies between access behaviors. Therefore, SEAM designs a structural modulation factor $S_{ij} \in R$ to describe the structural relevance between positions i and j in the knowledge graph. The final structure-enhanced attention weight is defined as:

$$\overline{a}_{ij} = \frac{\exp(Q_i \cdot K_j^T + \lambda S_{ij})}{\sum_{k=1}^{n} \exp(Q_i \cdot K_k^T + \lambda S_{ik}))}$$

Where λ is the importance coefficient for adjusting the structural weight, which is used to balance the influence of structural information and content information in attention calculation.

In terms of obtaining structural information, SEAM relies on the context subgraph representation extracted from the knowledge graph. For each access event x_i, its structural adjacency set is defined as $N(i)$, and its structural representation is:

$$s_i = \sum_{j \in N(i)} \phi(r_{ij}, x_j)$$

Where r_{ij} represents the semantic relationship type of the edge from i to j, and $\phi(\cdot)$ is a relationship-aware fusion function that is used to fuse neighbor features and edge semantic type encoding into a unified representation. Finally, the structural modulation factor S_{ij} can be approximated by the similarity or attention score between structural embeddings, for example:

$$S_{ij} = \varphi(s_i, s_j) = \frac{s_i^T s_j}{\| s_i \| \cdot \| s_j \|}$$

This structural enhancement mechanism makes the attention distribution not only limited to surface similarities but also introduces the relationship priors between entities in the semantic graph, thereby having a stronger ability to distinguish access patterns with similar semantics but different behaviors.

Finally, SEAM obtains context-aware behavior embedding representation by applying structure-enhanced attention to value vector aggregation:

$$\tilde{x}_i = \sum_{j=1}^{n} a_{ij} V_j$$

This embedding not only captures the long-distance semantic dependencies between access events but also integrates the contextual information implicit in the graph structure. Through this mechanism, the model can build a more structure-recognizing temporal behavior representation in complex cloud environments, providing a solid foundation for subsequent anomaly identification and semantic interpretation.

3.2 Dynamic Semantic Knowledge Graph

Dynamic Semantic Knowledge Graph (DSKG) aims to model complex, high-dimensional, and dynamically evolving entity relationships in cloud environments to provide structured semantic support and assist in the deep understanding of the access behavior context in abnormal identification tasks. Memory access behavior in cloud platforms often depends not only on the current task itself, but also on a variety of contextual factors such as tenant strategies, service topology, and scheduling decisions. There are clear but dynamically changing semantic relationships between these factors, so it is necessary to build a knowledge graph structure with time perception and semantic generalization capabilities. DSKG achieves unified abstraction

and structural encoding of multiple types of behavior backgrounds through entity-relationship-time triple modeling, providing context enhancement capabilities for behavior modeling and identification. Its module architecture is shown in Figure 3.

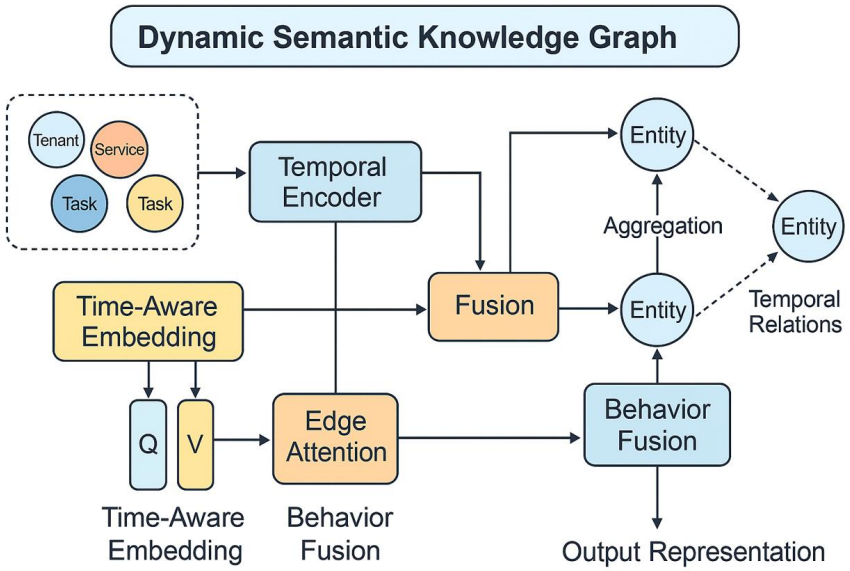


Figure 3. DSKG module architecture

Formally, the knowledge graph is defined as a set of triples $G = \{(h, r, t, \tau)\}$, where h and t represent the head entity and the tail entity, r represents the semantic relationship between them and τ represents the timestamp of the relationship. The entity space ε in the graph contains different types of system objects such as tenants, services, tasks, and nodes, while the relationship space R includes directed edges with contextual meanings such as dependencies, deployments, triggers, and shares. For each edge, a time-enhanced graph embedding is constructed:

$$e_{(h,r,t,\tau)} = f_{temporal}(e_h, e_r, e_t, \tau)$$

Where $f_{temporal}(\cdot)$ is the temporal encoding function, which integrates the graph structure and time evolution information.

To effectively aggregate the graph structure, the adjacency context set $N(v)$ is introduced to model the local subgraph of each entity node and calculate its structural context representation:

$$c_v = \sum_{(u,r,\tau)\in N(v)} a_{uvr} \cdot \phi(e_u, e_r, \tau)$$

Where a_{uvr} is the structural attention weight, which indicates the importance of the relationship between entities u and v in relation r, and $\phi(\cdot)$ is the relation-time-aware adjacency fusion function. This structural context is used to enhance the semantic context perception of access behavior.

At the full-graph level, to further construct a fusion representation across entity types, a graph representation fusion strategy is adopted to uniformly encode subgraph embeddings of different types, expressed as:

$$g_v = Fusion(e_v, c_v)$$

The Fusion module can aggregate the structure representation and the original entity embedding using a gating mechanism or an attention method, and output the final semantically enhanced node representation. These node embeddings can further form the global representation of the dynamic semantic graph:

$$G_{DSKG} = \{g_v \mid v \in \varepsilon\}$$

Through this modeling approach, DSKG can dynamically characterize the potential causal paths and semantic dependencies between different behaviors in the system, and provide structural constraints and contextual interpretation capabilities for the discriminant module.

Finally, in order to integrate the graph structure with the temporal behavior, the entity representation output by DSKG is fed into the fusion module corresponding to the access sequence embedding to construct the structure-aware behavior vector. This process can be formalized as:

$$\hat{x}_i = \gamma(x_i, g_i)$$

x_i is the original representation of the i-th element in the access sequence, g_i is the corresponding graph semantic embedding, and $\gamma(\cdot)$ is the fusion function (such as splicing, gating, multiplication, etc.). Through this mechanism, the model acquires the ability to jointly model the entity semantics and structural context behind each access behavior, providing a solid graph semantic foundation for the accurate identification of abnormal behavior.

4. Experimental Results

4.1 Dataset

This study uses the Alibaba Cluster Trace 2018 as the primary experimental dataset to simulate and analyze memory access behavior in real cloud environments. The dataset was collected by Alibaba Cloud from production-level clusters. It includes records of scheduling, resource allocation, and task execution over an extended period on thousands of servers. It has been widely adopted in large-scale cloud system modeling and behavior analysis, offering high representativeness and practical value.

The dataset provides detailed operational information at both the job and instance levels. It includes multidimensional metrics such as CPU, memory, disk usage, task lifecycles, job dependencies, and container scheduling. In this study, particular attention is paid to the instance-level resource usage logs and scheduling records. By modeling memory allocation and access states of tasks over time, high-dimensional time-series sequences and structural contexts are constructed. These serve as input features for the subsequent anomaly detection tasks.

In addition, the dataset contains mixed workloads across multiple tenants and hybrid scheduling patterns that involve both online and offline task executions. This makes it especially suitable for modeling complex semantic structures and multi-source dependencies in real-world scenarios. By building dynamic knowledge graphs and temporal behavior embeddings, it becomes possible to reconstruct the operational context of real cloud systems. This enhances the robustness and generalization capability of the model in practical settings.

4.2 Experimental setup

The experiments in this study were conducted on a high-performance computing node. The node is equipped with dual Intel Xeon Gold 6226R processors, providing a total of 48 cores. It also includes 512 GB of DDR4 memory, two NVIDIA A100 GPUs with 40 GB each, and a 1 TB NVMe SSD. This configuration supports large-scale graph embedding and high-dimensional time-series modeling tasks. All compute-

intensive modules, including graph neural network construction, attention weight training, and structure fusion, were executed on GPUs to significantly accelerate both training and inference.

The software environment was based on Ubuntu 20.04 LTS. Python 3.10 was used as the main programming language. PyTorch 2.1 was selected as the deep learning framework. DGL (Deep Graph Library) version 1.1 was used for handling dynamic graph structures and graph neural network training. Additional libraries such as NumPy, Pandas, matplotlib, and Scikit-learn were used for data preprocessing, visualization, and result analysis. GPU acceleration was supported by CUDA 12.1 and cuDNN 8.9 to ensure efficiency and stability during graph construction and attention computation.

During model deployment and training, a mixed precision training strategy was applied to optimize memory usage. Distributed tensor parallelism was used to support large-scale graph training. All model training was conducted with fixed random seeds to ensure reproducibility. Logging and monitoring were implemented using TensorBoard and Prometheus, integrated with Grafana. This setup enabled effective tracking and analysis of key performance metrics. The overall environment ensured stable operation under resource-intensive conditions and reliable experimental results.

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

Method	**F1-Score**	**AUC-ROC**	**Accuracy**
GraphSAGE[23]	78.2%	83.5%	80.1%
GAT[24]	81.4%	86.2%	82.7%
STG-NC[25]	84.7%	89.8%	85.5%
Ours	88.9%	93.1%	89.4%

As shown in the results of Table 1, the proposed model significantly outperforms all baseline methods in terms of F1-Score, AUC-ROC, and Accuracy. This demonstrates its strong performance in detecting abnormal memory access behavior in cloud environments. The performance advantage reflects the effectiveness of the proposed method in jointly modeling high-dimensional time-series data and complex structural semantics. It also shows the model's robustness under multi-tenant mixed workloads and dynamic resource scheduling. Compared with traditional graph neural network models, the proposed model more precisely captures contextual differences and semantic heterogeneity associated with abnormal behaviors.

Specifically, GraphSAGE lacks a fine-grained structural attention mechanism. It extracts information only from adjacent nodes in a static graph. This limits its classification accuracy when dealing with time-sensitive and frequently evolving access sequences. GAT introduces attention between nodes and assigns dynamic weights to neighbors. However, it still fails to model semantic paths and structural dependencies underlying access behaviors. As a result, its generalization ability remains limited in complex scenarios. STG-NC incorporates temporal modeling capabilities. Yet its structural representation relies on a fixed graph topology and lacks dynamic integration with contextual semantic graphs. This prevents it from capturing potential higher-order associations.

In contrast, the SEAM-DSKG model proposed in this study introduces a structure-enhanced attention mechanism and a dynamic semantic knowledge graph. It deeply integrates access behaviors with their semantic background. The model can not only detect surface-level behavioral changes but also understand the structural intent and entity associations behind these behaviors. This context-aware capability is crucial in cloud platforms. Many abnormal accesses are not isolated events but are triggered by structural or policy-level changes that lead to cascading effects. As a result, the model achieves leading performance across all metrics. This validates the practicality and advancement of combining structural and semantic modeling for cloud security classification tasks.

Furthermore, the improvement in AUC-ROC highlights the model's superior ability to distinguish the boundary between normal and abnormal behavior distributions. This indicates that the model not only performs accurate classification but also provides more reliable confidence estimates for anomalies. In real-world cloud platforms, this feature helps reduce both false positives and false negatives. It improves system response efficiency and enhances the accuracy of security policy adjustments. This also provides a solid data foundation for automated defense mechanisms and operational decisions. Overall, the experimental results confirm the practical value and engineering potential of the proposed method in dynamic, high-dimensional, and structurally complex cloud environments.

2) Ablation Experiment Results

This paper further gives the results of the ablation experiment as shown in Table 2.

Table 2: Ablation Experiment Results

Method	**F1-Score**	**AUC-ROC**	**Accuracy**
Baseline	81.0%	85.2%	82.3%
+SEAM	85.3%	89.6%	86.1%
+DSKG	84.5%	88.9%	85.2%
Ours	88.9%	93.1%	89.4%

As shown in the ablation results in Table 2, both core components of the proposed model—the Structure-Enhanced Attention Mechanism (SEAM) and the Dynamic Semantic Knowledge Graph (DSKG)—contribute significantly to the overall performance improvement. The baseline model, used as a reference, does not include structural information or contextual semantic modeling. Although it has some time-series modeling capability, its effectiveness in identifying abnormal behaviors in complex cloud environments is limited. In multi-tenant and multi-service scenarios, it shows a relatively high false positive rate.

When the SEAM module is added to the baseline, the model shows notable improvements across all three metrics. In particular, the F1-Score increases by 4.3 percentage points. This indicates that the structure-enhanced attention mechanism improves the model's ability to capture contextual dependencies between access behaviors. It enables the model to handle scenarios with complex behavior chains and long dependency paths more effectively. This structural awareness is critical for detecting abnormal access patterns hidden within normal scheduling processes, especially those that cannot be easily distinguished by surface features.

Similarly, adding the DSKG module also leads to stable performance gains. This is mainly due to the ability of the knowledge graph to model semantic relationships among multiple entities. DSKG enables the model to understand higher-level semantic factors behind access behaviors, such as tenant policies, service deployment locations, and task dependencies. As a result, the model gains a more comprehensive understanding of

behavioral context. Even when feature-level changes are subtle, the model can make reasonable judgments based on structural context. This helps reduce both false positives and false negatives.

The full model, which integrates both SEAM and DSKG, achieves the best results on F1-Score, AUC-ROC, and Accuracy. This demonstrates the synergistic effect of structural modeling and semantic enhancement. These results confirm that the proposed framework offers stronger classification and generalization capabilities in high-dimensional, dynamic, and structurally complex cloud environments. The joint modeling strategy provides an effective path toward building more robust and intelligent cloud security behavior detection systems.

3) Analysis of the impact of different numbers of attention heads on model performance

This paper further analyzes the impact of different numbers of attention heads on model performance, and the experimental results are shown in Figure 4.

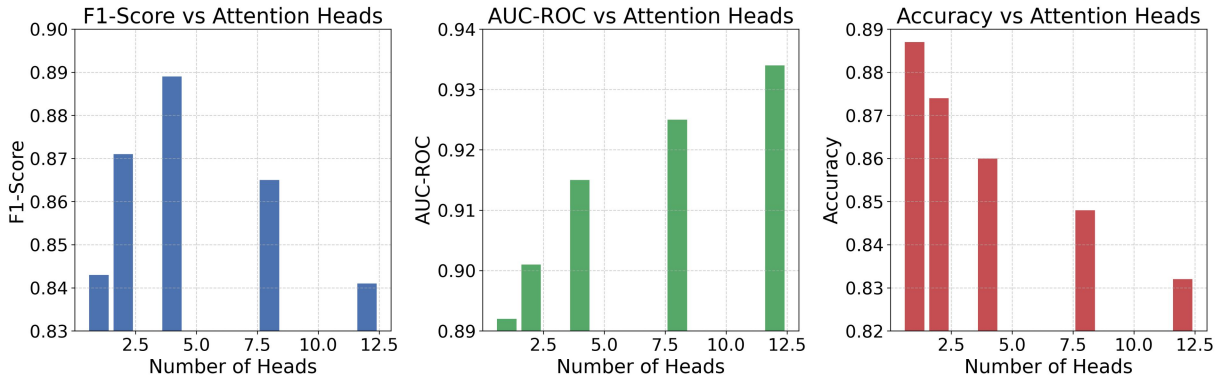


Figure 4. Analysis of the impact of different numbers of attention heads on model performance

Figure 4 shows that different numbers of attention heads influence the model in a clear yet nonlinear manner on three key metrics (F1-Score, AUC-ROC, and Accuracy). The effect indicates that the benefit of the Structure-Enhanced Attention Mechanism (SEAM) depends not only on using attention but also on setting its internal parameters. The headcount is a sensitive hyperparameter that strongly shapes final performance.

The F1-Score reaches its peak when the headcount is four. The curve is parabolic: too few heads limit modeling power, while too many add noise. A moderate head count improves the model's ability to outline behavior boundaries in high-dimensional access sequences and contextual graph structures. An excessive number of heads may dilute contextual information and weaken local semantic focus.

The AUC-ROC rises steadily as the headcount increases. More heads help the model draw a sharper confidence boundary between normal and abnormal cases. This suggests that multi-head attention improves classification robustness and structural generalization. The benefit is crucial in cloud environments where dynamic graph relations are unstable and multiple interaction paths must be captured.

Accuracy, in contrast, declines when the head count becomes large. The drop may result from overfitting or higher sensitivity to distribution shifts under high-head settings. The fluctuation shows that overall accuracy alone cannot reflect the real effectiveness of fine-grained behavior classification. In this task, the more stable AUC and F1 provide better guidance. Overall, the experiment confirms that the proposed model is sensitive to attention-related hyperparameters and offers clear directions for parameter tuning during deployment.

4) Comparison of discrimination effects under different knowledge graph update frequencies

This paper also gives a comparison of the discrimination effects under different knowledge graph update frequencies, and the experimental results are shown in Figure 5.

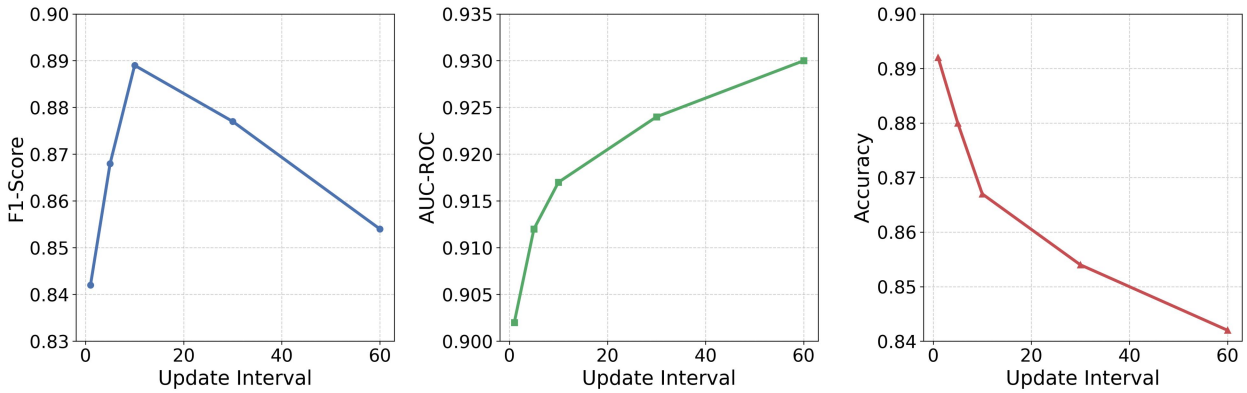


Figure 5. Comparison of discrimination effects under different knowledge graph update frequencies

The experimental results show that the update frequency of the knowledge graph affects the model in different ways. The F1-Score reaches its maximum when the update interval is 10 minutes, then decreases as the interval grows. AUC-ROC rises as the interval increases. Accuracy follows the opposite trend and declines slowly. These divergent patterns indicate that both very frequent and very sparse updates can impair performance. They also highlight the critical role of graph freshness in cloud security tasks.

The parabolic shape of the F1-Score curve can be explained as follows. A very high update rate, such as one minute, injects many short-term fluctuations into the graph structure and brings noise when the model tries to recover abnormal behaviors. A very low rate, such as 60 minutes, lets the graph fall behind real events and weakens precision. A moderate interval of 10 minutes balances timeliness and stability. It helps SEAM-DSKG build a more reliable link between continuous access sequences and dynamic context. This leads to the best overall classification of anomalies.

The upward trend of AUC-ROC with longer intervals shows that ranking ability improves even if overall accuracy falls slightly. Fewer updates reduce structural jitter during graph reconstruction. The model can then measure normal and abnormal distances with a more consistent semantic view. When decision thresholds are tunable or use adaptive strategies, a lower update frequency can raise confidence in the predictions. This matches real cloud deployments where threshold optimization is common.

The steady decline in accuracy reveals that a static threshold is sensitive to different levels of graph freshness. A longer interval lets the model accept some samples that are normal in outdated semantics, which lowers the hit rate. If a fixed threshold is used for hard classification, a long interval may weaken real-time alerts. Dynamic thresholds or cost-sensitive strategies could offset this effect. Overall, the results stress the coupling between the graph update strategy and threshold design. Practical deployments must weigh update cost, timeliness, and tolerance mechanisms to gain the full benefit of dynamic semantic graphs in anomaly detection.

5) Sensitivity analysis of temporal coding dimension to semantic enhancement effect

This paper also gives a sensitivity analysis of the time coding dimension on the semantic enhancement effect, and the experimental results are shown in Figure 6.

Figure 6 shows the effect of different temporal encoding dimensions on the model's F1-Score. The results indicate a clear nonlinear relationship in the role of temporal encoding within semantic enhancement. When the encoding dimension is low, such as 4 or 8, the model has limited capacity to represent temporal dynamics. This weakens its ability to capture time-related features in behavior sequences, which affects the performance of context semantics in the structure-enhanced attention mechanism.

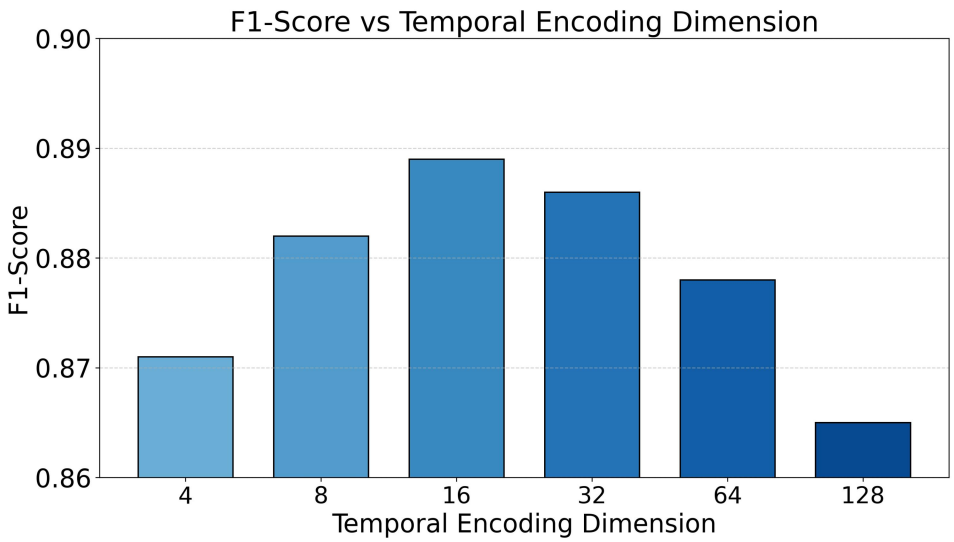


Figure 6. Sensitivity analysis of temporal coding dimension to semantic enhancement effect

As the temporal dimension increases, the model achieves optimal performance around 16 dimensions. This suggests that, at this level, temporal information is well integrated with graph-based semantics. It enhances the model's ability to distinguish abnormal access behaviors. However, when the dimension increases further to 64 or 128, the performance declines. This may be due to excessive dimensionality introducing redundant noise. The representation space becomes sparse, which interferes with learning a clear decision boundary.

These results show that, when building dynamic knowledge graphs, temporal embedding is as important as the graph structure itself. The granularity and compression of time representations are critical. If the granularity is too coarse, the model cannot reflect fine-grained behavior rhythms. If too fine, it increases complexity without adding meaningful information. Therefore, in semantic enhancement tasks, the temporal encoding dimension must be carefully tuned. It should match the coupling between behavior sequences and contextual structure to achieve optimal classification performance.

5. Conclusion

This paper addresses the problem of complex memory access behavior classification in cloud computing environments. It proposes a high-dimensional time-series modeling framework that combines a structure-enhanced attention mechanism with a dynamic semantic knowledge graph. The approach jointly models access sequences and their contextual structural semantics. It improves the model's ability to represent abnormal behaviors and enhances classification robustness. By integrating attention with graph structures, the model captures fine-grained dependencies across tasks and tenants. It also demonstrates higher accuracy and interpretability than traditional methods in dynamic, multi-tenant cloud settings.

At the methodological level, the paper introduces the Structure-Enhanced Attention Mechanism (SEAM). This enhances the model's ability to perceive structural dependencies within behavior patterns. Meanwhile, the construction of the Dynamic Semantic Knowledge Graph (DSKG) allows the model to continuously track semantic changes in system states. This addresses the limitations of static graph modeling, which often lags behind and becomes unstable. Through comparative experiments, ablation studies, and sensitivity analysis, the paper systematically validates the effectiveness of each module under real cloud data. It further confirms the critical role of semantic enhancement in behavior classification tasks.

From an application perspective, the results of this study can support traditional cloud intrusion detection and resource misuse monitoring. They can also extend to emerging areas such as container security, edge-

intelligent collaborative detection, and anomaly detection in microservice dependencies. In scenarios with strong isolation, high elasticity, and multiple tenants, the combined modeling of structure and time enables more accurate risk identification and dynamic policy scheduling. This provides a methodological foundation for building intelligent cloud security systems with real-time responsiveness. The proposed framework is also scalable and can be deployed in production-grade monitoring systems, offering clear engineering value.

Future research can proceed in several directions. One challenge is the automatic evolution of graph structures. Self-supervised graph learning and generative graph construction could improve graph representation quality. Another issue is resource efficiency in training and inference under large-scale distributed systems. Lightweight architectures and online incremental updates may be explored. In practical deployments, integrating risk assessment and interpretability frameworks could further promote the model's application in high-security domains such as critical infrastructure, financial clouds, and government clouds. In conclusion, the proposed modeling approach provides a solid foundation for advancing intelligent security in cloud computing. It also offers a general solution for high-dimensional and heterogeneous behavior modeling tasks.

References

[1] Qureshi K N, Jeon G, Piccialli F. Anomaly detection and trust authority in artificial intelligence and cloud computing[J]. Computer Networks, 2021, 184: 107647.

[2] Vervaet A. Monilog: An automated log-based anomaly detection system for cloud computing infrastructures[C]//2021 IEEE 37th international conference on data engineering (ICDE). IEEE, 2021: 2739-2743.

[3] Moreira D A B, Marques H P, Costa W L, et al. Anomaly detection in smart environments using AI over fog and cloud computing[C]//2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2021: 1-2.

[4] Abdallah A M, Alkaabi A S R O, Alameri G B N D, et al. Cloud network anomaly detection using machine and deep learning techniques—Recent research advancements[J]. IEEE Access, 2024, 12: 56749-56773.

[5] Gudelli V R. Anomaly detection in cloud networks using machine learning algorithms[J]. African Journal of Artificial Intelligence and Sustainable Development, 2024, 4(1).

[6] Yu X, Yang X, Tan Q, et al. An edge computing based anomaly detection method in IoT industrial sustainability[J]. Applied Soft Computing, 2022, 128: 109486.

[7] Devi T A, Jain A. Enhancing Cloud Security with Deep Learning-Based Intrusion Detection in Cloud Computing Environments[C]//2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT). IEEE, 2024: 541-546.

[8] Girish L, Rao S K N. Anomaly detection in cloud environment using artificial intelligence techniques[J]. Computing, 2023, 105(3): 675-688.

[9] Lei X, Xia Y, Wang A, et al. Mutual information based anomaly detection of monitoring data with attention mechanism and residual learning[J]. Mechanical Systems and Signal Processing, 2023, 182: 109607.

[10] Sun H, Chen M, Weng J, et al. Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism[J]. IEEE Transactions on Vehicular Technology, 2021, 70(10): 10880-10893.

[11] Hu W, Cao L, Ruan Q, et al. Research on anomaly network detection based on self-attention mechanism[J]. Sensors, 2023, 23(11): 5059.

[12] Li, L., Liu, Z., Chen, C., Zhang, Y. L., Zhou, J., & Li, X. (2019). A time attention based fraud transaction detection framework. arXiv preprint arXiv:1912.11760.

[13] Li, J., Xu, C., Feng, B., & Zhao, H. (2023). Credit risk prediction model for listed companies based on CNN-LSTM and attention mechanism. Electronics, 12(7), 1643.

[14] Xiang L, Yang X, Hu A, et al. Condition monitoring and anomaly detection of wind turbine based on cascaded and bidirectional deep learning networks[J]. Applied Energy, 2022, 305: 117925.

[15]Zhong L, Wu J, Li Q, et al. A comprehensive survey on automatic knowledge graph construction[J]. ACM Computing Surveys, 2023, 56(4): 1-62.

[16]Peng C, Xia F, Naseriparsa M, et al. Knowledge graphs: Opportunities and challenges[J]. Artificial Intelligence Review, 2023, 56(11): 13071-13102.

[17]Yang Y, Huang C, Xia L, et al. Knowledge graph contrastive learning for recommendation[C]//Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. 2022: 1434-1443.

[18]Zhu X, Li Z, Wang X, et al. Multi-modal knowledge graph construction and application: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 36(2): 715-735.

[19]Zeng X, Tu X, Liu Y, et al. Toward better drug discovery with knowledge graph[J]. Current opinion in structural biology, 2022, 72: 114-126.

[20]Chandak P, Huang K, Zitnik M. Building a knowledge graph to enable precision medicine[J]. Scientific Data, 2023, 10(1): 67.

[21]Santos A, Colaço A R, Nielsen A B, et al. A knowledge graph to interpret clinical proteomics data[J]. Nature biotechnology, 2022, 40(5): 692-702.

[22]Liu, Z., Chen, C., Yang, X., Zhou, J., Li, X., & Song, L. (2018, October). Heterogeneous graph neural networks for malicious account detection. In Proceedings of the 27th ACM international conference on information and knowledge management (pp. 2077-2085).

[23]Huang K, Chen C. Subgraph generation applied in GraphSAGE deal with imbalanced node classification[J]. Soft Computing, 2024, 28(17): 10727-10740.

[24]Guo S. A Survey on GAT-like Graph Neural Networks[C]//2020 International Conference on Communications, Information System and Computer Engineering (CISCE). IEEE, 2020: 303-308.

[25]Choi, J., Choi, H., Hwang, J., & Park, N. (2022, June). Graph neural controlled differential equations for traffic forecasting. In Proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 6, pp. 6367-6374).