Transactions on Computational and Scientific Methods | Vol. 4, No. 12, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Graph Neural Recognition of Malicious User Patterns in Cloud Systems via Attention Optimization

Danyi Gao
Columbia University, New York, USA
dg3224@columbia.edu

Abstract: To address the growing issue of malicious user behavior in cloud computing environments, this paper proposes a recognition algorithm based on an Improved Graph Attention Network (IGAT). The method leverages the structural modeling capability of graph neural networks for behavioral data. By introducing a Multi-Scale Neighbor Attention mechanism and a Context-Aware Attention Adjustment strategy, the model improves its ability to represent hidden abnormal relationships in user behavior graphs. The Multi-Scale Neighbor Attention mechanism builds neighbor information across different hop ranges. This enhances the global awareness of node representations. The Context-Aware Attention Adjustment strategy uses historical behaviors and interaction context to dynamically refine the original attention distribution. This improves the model's ability to capture complex behavioral semantics. The public UNSW-NB15 security dataset is used to construct user behavior graphs. A series of comparative and ablation experiments is conducted to evaluate the model. The proposed method demonstrates superior performance in terms of accuracy, F1-Score, and AUC. It is also tested under different attack types, node densities, and model complexities to assess its stability and efficiency. Experimental results show that the model offers strong detection ability and robustness. It also maintains good inference performance under resource constraints. These findings demonstrate their practical value for large-scale cloud platforms.

Keywords: graph neural network; malicious behavior recognition; attention mechanism; user behavior graph

1. Introduction

With the rapid development of information technology, cloud computing has become a key driver of digital transformation. Its features, such as on-demand allocation, high elasticity, low cost, and strong scalability, have led more organizations and individuals to migrate critical data and applications to cloud platforms[1,2]. However, as cloud computing becomes more widely adopted, security issues have become increasingly prominent. In particular, the concealment and destructiveness of malicious user behavior in the cloud pose serious challenges to platform security. Traditional security measures are often ineffective in identifying complex and variable malicious behavior patterns. There is an urgent need to leverage artificial intelligence for more intelligent security protection[3].

User behavior in cloud environments is characterized by multi-source heterogeneity, high dimensionality, and strong temporal dependency, which makes malicious behavior detection extremely challenging. On one hand, users interact frequently, forming complex relational networks. On the other hand, malicious users often adopt disguise strategies, making their behavior patterns highly similar to those of legitimate users. This enables them to evade traditional detection mechanisms. In this context, it is essential to explore the deep

structural relationships in behavioral data and to build efficient and accurate identification models. This is not only a valuable supplement to traditional security systems but also a key direction for advancing intelligent security[4].

Graph Neural Networks have emerged as effective methods for handling graph-structured data due to their strong relational modeling capabilities. Among them, the Graph Attention Network (GAT) introduces an attention mechanism to differentially model neighbor information, improving its expressive power on complex graph data. This makes GAT widely applicable in user behavior modeling and social network analysis[5]. However, the original GAT algorithm still faces limitations when applied to large-scale user behavior graphs in cloud environments. These include limited expressive ability, low computational efficiency, and high sensitivity to noise. Therefore, improvements are necessary to better meet the needs of cloud-based malicious behavior detection[6].

By improving the GAT algorithm, the model can better capture complex user relationships and enhance the accuracy of malicious behavior detection. For example, introducing multi-scale attention mechanisms can more comprehensively capture multi-level neighbor relationships. Optimizing weight calculation strategies can enhance sensitivity to abnormal patterns. Incorporating contextual information to dynamically adjust neighbor weights can improve the modeling of behavior evolution trends. These improvements help distinguish normal from abnormal user behaviors more accurately, enhance system security, reduce potential threat losses, and increase the trustworthiness of cloud services[7].

This study focuses on malicious user behavior detection in cloud environments based on an improved GAT algorithm. It aims to develop an intelligent security detection framework suitable for complex cloud environments. In real-world scenarios, both cloud service providers and users urgently need an efficient and scalable detection technology to cope with evolving security threats. This study holds significant theoretical and technical value and shows strong potential for practical application. Exploring the deep application of graph neural networks in the security domain can provide new approaches and technical support for user behavior modeling and security defense in cloud computing. It lays a solid foundation for building a more intelligent and reliable cloud security ecosystem.

2. Related work

2.1 Graph Neural Networks

Graph Neural Networks (GNNs) have gained wide attention in recent years as deep learning models designed for graph-structured data. Unlike traditional neural networks that mainly process data in Euclidean space, GNNs can effectively model complex relationships in non-Euclidean structures. They learn node representations by aggregating information from neighboring nodes[8,9]. This capability gives GNNs a significant advantage in applications such as social networks, recommendation systems, and knowledge graphs. Basic GNN models use static neighbor aggregation to propagate features. However, their expressive power is limited when applied to real-world tasks with diverse structures and dynamic evolution. In particular, their performance is restricted when dealing with high-dimensional, heterogeneous graph data[10].

To overcome these limitations, researchers have proposed various improved GNN architectures. Among them, the Graph Attention Network (GAT) introduces an attention mechanism into basic graph convolution. This allows the model to assign different weights to neighbor nodes based on their importance. Such differentiated information processing enhances the model's expressive power and improves its robustness to noise and irrelevant data[11]. The development of GAT has improved GNN performance in tasks such as node classification and link prediction. However, the original GAT still faces challenges in handling large-scale graphs or high-frequency interaction data. It struggles with efficiency and generalization, especially in

scenarios requiring long-range dependency modeling or dynamic structural adaptation. Optimizing the structure and mechanism of GAT has therefore become an important research focus in recent years[12,13].

In cloud computing environments, user relationships are complex and change frequently. Behavioral data often exhibit high dimensionality, heterogeneity, and temporal characteristics[14,15]. These features place higher demands on the modeling ability of graph neural networks. Improving learning efficiency on large-scale graphs and enhancing the detection of rare abnormal behaviors are key issues when applying GNNs to cloud security tasks. To address this, many studies have extended and optimized GAT from different aspects, such as attention mechanisms, graph structure representation, and dynamic update strategies. These efforts aim to better capture the latent structural information and temporal evolution in graphs. Such improvements enhance the model's adaptability and discriminative ability in complex environments[16]. This research provides a solid technical foundation for applying GNNs to security tasks like malicious behavior detection, and offers valuable approaches for future studies.

2.2 Malicious user detection

Malicious user detection is a key research direction in the field of cybersecurity. Its importance becomes even more prominent in cloud computing environments[17,18]. As cloud platforms become more widely used, user interactions grow more frequent and complex[19,20]. This creates favorable conditions for malicious users to hide their behavior. These users often exploit the openness and elasticity of cloud resources to carry out data theft, privilege escalation, and denial-of-service attacks. Such activities pose serious threats to system security[21]. Traditional detection methods rely heavily on rule matching, feature engineering, and statistical models. Although effective in certain scenarios, they often fail to detect well-concealed attacks in highly dynamic and large-scale cloud environments[22,23]. They also suffer from poor generalization and exhibit high false positive and false negative rates[24].

In recent years, the development of machine learning and deep learning has led to a shift toward data-driven models for malicious user detection. These approaches typically use user behavior data to extract features and train classification models. The goal is to distinguish between normal and malicious users automatically[25]. In cloud platforms, behavioral logs, resource access records, and session histories provide rich data for intelligent detection. However, two major challenges remain. First, it is difficult to select key features with strong discriminative power from high-dimensional data. Second, it is challenging to model the complex interactions between users to improve the detection of organized and collaborative attacks. Relying solely on flat features or time series analysis is not sufficient to capture hidden patterns and relational structures. This limits the accuracy of detection models.

Graph-based modeling offers a new breakthrough for malicious user detection. By representing user behaviors and interactions as a graph, it becomes possible to uncover hidden relationships and improve the detection of collaborative malicious actions[26]. On this basis, introducing Graph Neural Networks, especially attention-based models, provides more powerful tools for behavior discrimination. GNNs can effectively integrate both structural information and node features. They also support multi-layer aggregation to learn higher-order relationships, improving both accuracy and robustness. With an improved GAT algorithm, the model can better capture subtle differences in behavior patterns. This makes it more suitable for complex and evolving malicious behavior detection in cloud environments. As a result, it offers a feasible solution for building intelligent and efficient cloud security systems.

3. Method

This study proposes a cloud-based malicious user behavior detection method based on an Improved Graph Attention Network (IGAT). The goal is to enhance the model's ability to represent complex user relationships and abnormal behavior patterns. To achieve this, the method introduces two key improvements to the original GAT. First, a Multi-Scale Neighbor Attention (MSNA) mechanism is designed. It enables

dynamic aggregation of neighboring node information across different structural levels. This improves the model's ability to integrate both local and global behavioral features. Second, a Context-Aware Attention Adjustment (CAAA) strategy is proposed. This mechanism uses user behavior history and interaction context to dynamically adjust the importance of neighboring nodes. It improves the model's sensitivity to potential malicious behavior. With these two improvements, IGAT can more effectively capture key discriminative features from complex behavior graphs. This provides a more accurate and generalizable modeling approach for detecting malicious users in cloud platforms. The architecture of the overall model is illustrated in Figure 1.

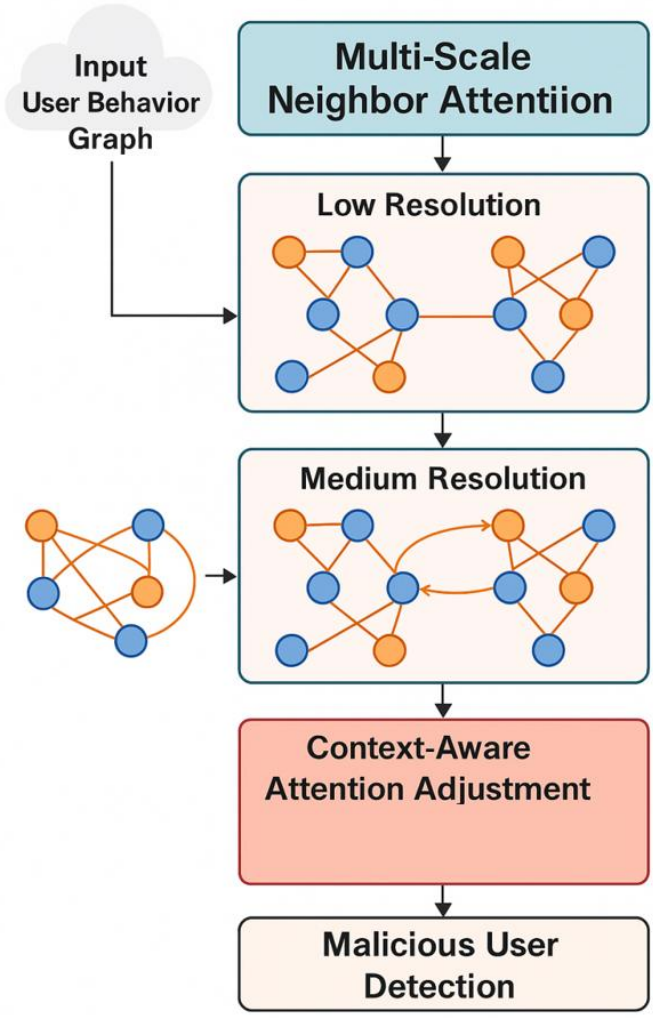


Figure 1. Overall model architecture diagram

3.1 Multi-Scale Neighbor Attention

In order to enhance the model's capacity to represent complex structural relationships, this study introduces a Multi-Scale Neighbor Attention mechanism (Multi-Scale Neighbor Attention, MSNA) built upon the original graph attention mechanism. The main objective of this mechanism is to capture relationship patterns between nodes at multiple levels, ranging from local to global, by constructing neighbor subgraphs at different hop distances. This approach effectively addresses the limitations of the original GAT in expressing hierarchical and wide-range structural dependencies. Specifically, the model independently

computes attention scores for neighbor nodes at each scale, based on their hop count relative to the target node. These multi-level attention outputs are then aggregated through a fusion process to form a richer semantic representation for each node. Through this design, the model preserves the strong influence of immediate neighbors while incorporating structural signals from more distant nodes. This leads to a more complete and nuanced understanding of the latent relational patterns embedded in the behavior graph. The detailed architecture and functional flow of the MSNA module are shown in Figure 2.

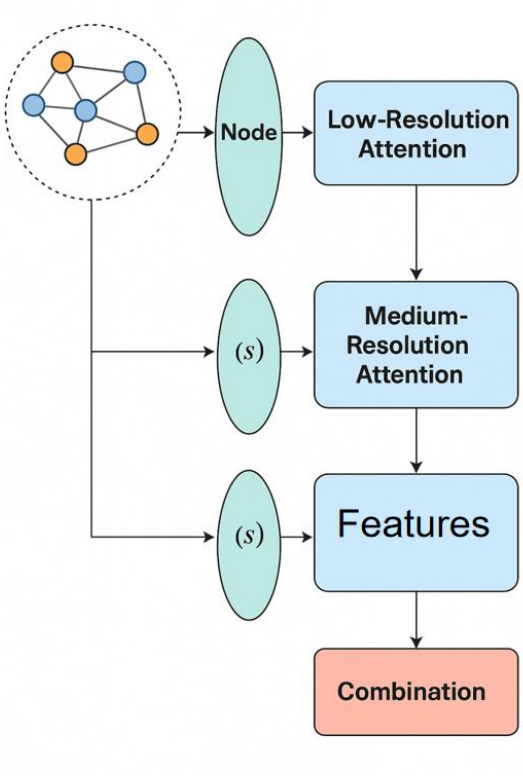


Figure 2. MSNA module architecture

At each scale s, the attention weight of node i and its s-hop neighbor set $N_i^{(s)}$ is first calculated as follows:

$$e_{ij}^{(s)} = LeakyRELU(a^{(s)^T}[W^{(s)}h_i \,||\, W^{(s)}h_j]), \quad j \in N_i^{(s)}$$

Among them, $W^{(s)}$ represents the linear transformation matrix at the s-th layer scale, $a^{(s)}$ is the learnable attention vector, and $||$ represents the concatenation operation. Then the attention weights of the neighboring nodes are obtained through Softmax normalization:

$$a_{ij}^{(s)} = \frac{\exp(e_{ij}^{(s)})}{\sum_{k \in N_i^{(s)}} \exp(e_{ik}^{(s)})}$$

The node representation at each scale is obtained by weighted aggregation:

$$h_i^{(s)} = \sigma\Big(\sum_{j \in N_i^{(s)}} a_{ij}^{(s)} W^{(s)} h_j\Big)$$

Where σ is a nonlinear activation function, such as ReLU or ELU. In order to fully integrate node information at different scales, a fusion function is used to perform a weighted combination of representations at each scale to form the final multi-scale representation:

$$h_i^{multi} = \sum_{s=1}^{S} \beta^{(s)} \cdot h_i^{(s)}$$

$\beta^{(s)}$ is the fusion weight of each scale, which can be automatically learned through training or using a predefined weight distribution. This mechanism effectively improves the model's ability to adapt to the diversity of graph structures and the complexity of behavioral patterns, and provides a more discriminative representation basis for subsequent attention adjustment.

3.2 Context-Aware Attention Adjustment

In order to further improve the model's ability to discriminate complex behavior patterns, this study introduces the context-aware attention adjustment mechanism (CAAA), which aims to enhance the ability to understand the behavioral context during the attention allocation process. Traditional GAT usually only considers the static features of the current node and its neighbors in attention calculation, ignoring the dynamic context and historical behavior information of the node in the graph. To make up for this deficiency, the CAAA mechanism, based on the original attention weight, fuses the contextual representation of the node to perform weight correction, thereby increasing sensitivity to potential abnormal relationships. Its module architecture is shown in Figure 3.

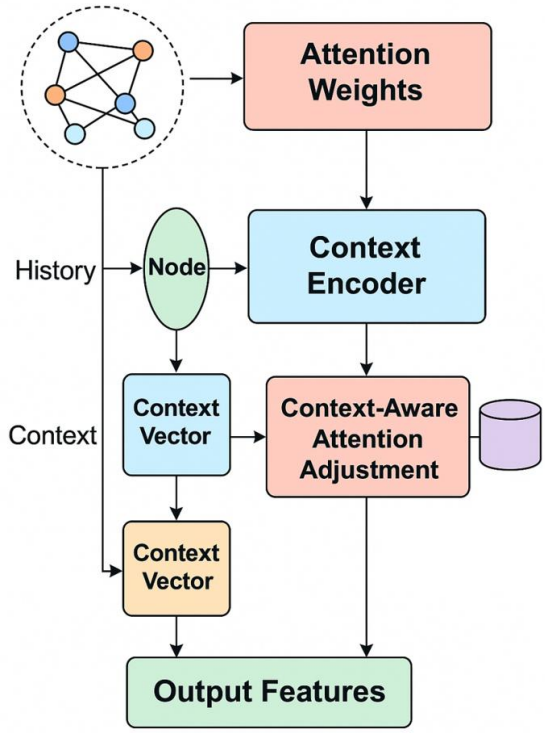


Figure 3. CAAA module architecture

Specifically, we first define the original attention weight as:

$$\alpha_{ij}^{(0)} = \frac{\exp(LeakyRELU(a^T[W^{(s)}h_i \| W^{(s)}h_j]))}{\sum_{k \in N_i} \exp(LeakyRELU(a^T[W^{(s)}h_i \| W^{(s)}h_j]))}$$

The context encoder $\phi(\cdot)$ is then introduced to obtain the context vector c_i of node i, which can come from the historical behavior sequence, interaction structure embedding, or path representation in the graph, etc:

$$c_i = \phi(H_i)$$

On this basis, a context-sensitive adjustment function $g(\cdot)$ is constructed to recalculate the attention coefficient:

$$\overline{a}_{ij} = g(a_{ij}^{(0)}, c_i, h_j)$$

A typical adjustment function can be expressed as weighted fusion:

$$\overline{a}_{ij} = \text{softmax}(a_{ij}^{(0)} + w^T \tanh(Uc_i + Vh_j))$$

Where A is a learnable parameter. Finally, the weighted aggregation of neighbor features is completed using the adjusted attention weights:

$$h'_i = \sigma(\sum_{j \in N_i} \overline{a}_{ij} \cdot Wh_j)$$

By introducing contextual factors for dynamic adjustment, the model can not only identify abnormal connections in static structures, but also perceive potential threats brought by behavioral evolution, significantly enhancing the ability to model complex attack behaviors and providing a more adaptive and discriminative representation for cloud malicious user detection tasks.

4. Experimental Results

4.1 Dataset

This study uses the UNSW-NB15 dataset as the experimental data source. The dataset consists of a mix of real network traffic and simulated attack behaviors. It is widely used in research related to cybersecurity and anomaly detection. By deploying various attack tools in a real network environment, the dataset generates comprehensive traffic data. It includes common malicious behaviors such as DoS, analysis, reconnaissance, and backdoor attacks. The dataset offers high diversity and realistic representation of attacks.

The dataset contains about 2.5 million network connection records. Each record includes 49 features. These features cover basic network attributes such as protocol, service port, and flow size, as well as content features and time-based statistical features. Each record is labeled with either normal behavior or a specific attack type. These labels provide a reliable source of supervision for learning models.

Compared with traditional datasets like KDD99, UNSW-NB15 is more aligned with modern network environments. It is more up-to-date in data composition, attack coverage, and feature structure. The dataset's rich user behavior patterns and malicious traffic interactions make it well-suited for graph construction. User behaviors can be modeled as nodes and their interactions as edges. This provides a solid data foundation for applying graph neural networks to malicious behavior detection tasks.

4.2 Experimental setup

In this study, a user behavior graph is constructed based on the UNSW-NB15 dataset. Each network connection record is abstracted as a node in the graph. Edges between nodes are created based on the same

IP address segment, similar communication patterns, or close time windows. This forms a behavior graph with structural features.

During the training of the graph neural network, 80 percent of the data is used as the training set, 10 percent as the validation set, and the remaining 10 percent as the test set. This ensures the model's generalization under different data distributions. All features are standardized before input to improve model convergence speed and avoid gradient instability caused by large differences in feature values. The graph neural network model is implemented using the PyTorch Geometric framework. The optimizer is Adam, with an initial learning rate of 0.001. The batch size is set to 64, and the number of training epochs is 200. The attention mechanism uses a multi-head structure with 8 attention heads per layer. The activation function is ELU. For the multi-scale structure, neighbor aggregation is performed at each scale. The final node representation is generated through a weighted fusion mechanism. The specific parameters are shown in Table 1.

Table 1: Specific parameter diagram

Parameter name	**Setting Value**
Optimizer	Adam
Learning Rate	0.001
Batch Size	64
Epochs	200
Attention Heads	8
Activation Function	ELU
Training / Val / Test Split	80% / 10% / 10%
Platform	PyTorch Geometric

4.3 Experimental Results

1) Comparative experimental results

This paper first gives the comparative experimental results, as shown in Table 2.

Table2: Comparative experimental results

Method	**Accuracy**	**AUC**	**F1-Score**
GCN[27]	89.7	87.3	88.4
GAT[28]	91.1	89.6	90.3
GraphSAGE[29]	90.5	88.2	89.1
GCN-LPA[30]	90.0	87.9	88.7
Ours	93.8	92.1	92.9

According to the results of the comparative experiments, the proposed Improved Graph Attention Network model (Ours) outperforms existing mainstream graph neural network methods on several key evaluation metrics. Specifically, it achieves 93.8 percent in Accuracy, 92.1 percent in AUC, and 92.9 percent in F1-Score. These results show comprehensive improvements in overall detection accuracy, classification stability, and the ability to capture malicious behavior. This demonstrates that the proposed structural enhancements

significantly improve the model's discriminative capacity. It also shows better generalization and robustness when handling complex cloud-based behavior data.

Compared with the traditional GCN method, the proposed approach introduces a Multi-Scale Neighbor Attention mechanism (MSNA) to address the limitations of single-scale structures in capturing long-range dependencies. Although GCN has some capability in modeling local neighbors, its fixed structure limits the learning of global graph information. This leads to weaker performance in AUC and F1-Score. The proposed method aggregates neighbor nodes at multiple hop distances and integrates the information through dynamic weighted fusion. This enables more effective separation of normal users from malicious users with hidden behaviors.

In addition, the Context-Aware Attention Adjustment mechanism (CAAA) also plays an important role as shown in the experimental results. A comparison between GAT and the proposed method reveals that although GAT includes an attention mechanism, its attention allocation remains limited to static structure and current features. By introducing context modeling, the proposed method encodes the historical behaviors and interaction context of nodes. This makes the attention weights better aligned with behavioral evolution patterns. It improves the accuracy and stability of detecting complex attack patterns, especially for ambiguous or borderline abnormal behaviors.

Overall, the experimental results validate the effectiveness of IGAT in malicious user detection tasks. They also confirm that structural enhancement and semantic adjustment work synergistically to improve the performance of graph-based classification models. By integrating structural information, behavioral context, and multi-scale aggregation mechanisms, the proposed method offers a practical solution to the challenge of malicious behavior detection in cloud environments. It also provides a scalable graph learning framework for future related research.

2) Ablation Experiment Results

This paper also further gives the results of the ablation experiment, and the experimental results are shown in Table 2.

Table 2: Ablation Experiment Results

Method	**Accuracy**	**AUC**	**F1-Score**
BaseLine	91.1	89.6	90.3
+MSNA	92.4	90.8	91.6
+CAAA	92.1	90.5	91.2
Ours	93.8	92.1	92.9

The results of the ablation experiments show that the two core improvement modules proposed in this study, the Multi-Scale Neighbor Attention mechanism (MSNA) and the Context-Aware Attention Adjustment mechanism (CAAA), both contribute significantly to the overall model performance. Based on the baseline model, which includes no structural enhancement or contextual adjustment, the Accuracy reaches 91.1 percent, the AUC is 89.6 percent, and the F1-Score is 90.3 percent. While the baseline model shows a certain level of detection capability, it still produces errors when dealing with complex or highly concealed malicious behaviors.

After introducing the MSNA module, model performance improves significantly. The Accuracy increases to 92.4 percent, and the F1-Score rises to 91.6 percent. This indicates that the multi-scale structure plays a key role in enhancing the model's ability to capture dependency relationships among neighbors at different levels in the graph. The mechanism effectively expands the model's receptive field. It helps capture multi-level

behavioral cues and improves the discrimination of potential malicious behavior paths. This is especially helpful for detecting coordinated group attacks in complex cloud environments.

Adding only the CAAA module also results in steady performance improvement. The AUC increases from 89.6 percent to 90.5 percent, and the F1-Score improves to 91.2 percent. These results indicate that incorporating contextual information into the dynamic adjustment of attention weights enhances the model's understanding of abnormal behavioral contexts. This allows previously hidden or vague behavioral patterns to be expressed more clearly. The improvement is especially evident when dealing with time-evolving data and subtle behavioral differences in cloud environments.

Finally, when both MSNA and CAAA modules are integrated into the model to form the complete IGAT framework, all performance metrics reach their highest values. The Accuracy reaches 93.8 percent, the AUC reaches 92.1 percent, and the F1-Score reaches 92.9 percent. These results confirm the synergistic effect of the two mechanisms in extracting structural information and understanding behavioral semantics. They demonstrate that the proposed method can complete the task of malicious user behavior detection more comprehensively and efficiently. This is particularly suitable for cloud environments with complex structures and diverse attack patterns.

3) Evaluation of the model's recognition ability under different attack types

This paper also provides a comprehensive evaluation of the model's recognition ability when applied to various categories of attack types commonly encountered in cloud computing environments. By constructing distinct subsets of the dataset corresponding to different malicious behavior patterns, the model's capability to differentiate and identify each specific attack type is thoroughly assessed. This process involves applying the proposed recognition framework across a range of predefined attack scenarios to observe how effectively it captures the structural and semantic differences inherent to each category. The aim of this evaluation is to demonstrate the flexibility and adaptability of the model when confronted with diverse and potentially imbalanced malicious traffic. The experimental configuration ensures consistency in feature input and model parameters across all attack types to isolate the impact of behavioral variance. The corresponding visualization and detailed outcomes of this evaluation process are illustrated in Figure 4.

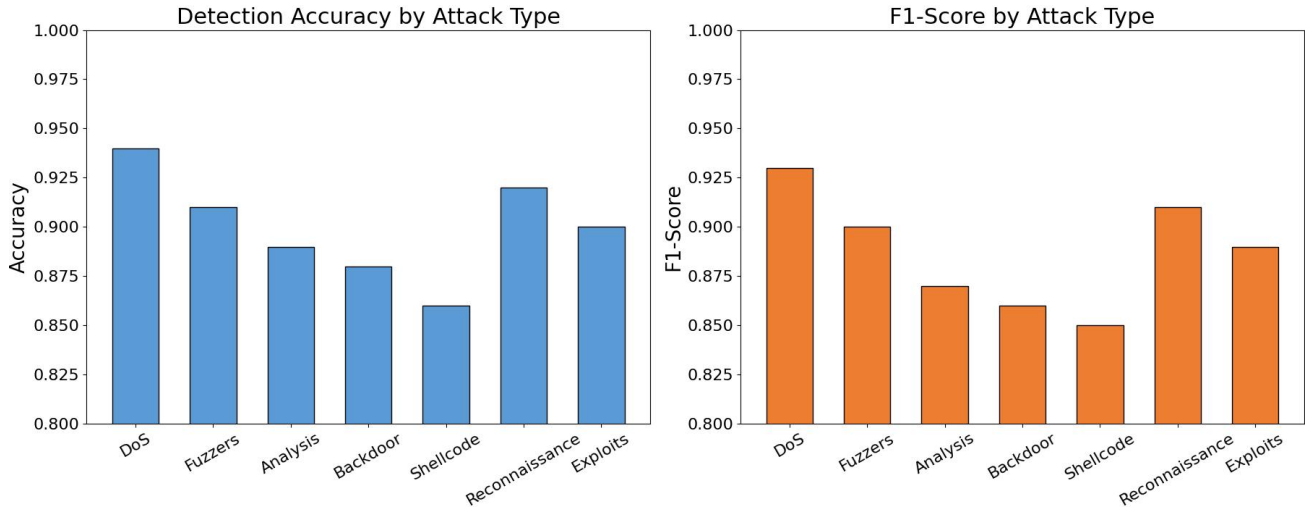


Figure 4. Evaluation of the model's recognition ability under different attack types

As shown in Figure 4, the proposed model exhibits varying recognition performance across different attack types, but the overall results remain stable. This indicates strong generalization ability. In the Accuracy chart, the model performs best on DoS and Reconnaissance attacks. The accuracy for both exceeds 0.92. This

suggests that the model can effectively capture the structural patterns of these attack types. In scenarios involving large volumes of access requests or scanning activities, the model responds with high accuracy.

In contrast, for more concealed and semantically complex attacks such as Shellcode and Backdoor, the accuracy and F1-Score are relatively lower. This difference indicates that attack types with weaker features or stronger context dependence pose greater challenges to the model. Even so, the Context-Aware Attention mechanism helps maintain the performance for these types above 0.85. This confirms the importance of semantic adjustment in detecting subtle abnormal signals.

The F1-Score chart further reflects the model's ability to balance precision and recall. The overall trend of F1-Score is consistent with the accuracy. DoS and Reconnaissance again show strong F1 performance. This means the model can detect most attack samples while minimizing false positives. The F1-Score for categories like Shellcode is slightly lower but remains within an acceptable range. This shows that the model can maintain stable performance even under class imbalance or ambiguous attack boundaries. Overall, the experimental results demonstrate the advantage of the multi-scale attention mechanism in modeling different levels of attack behaviors. They also highlight the complementary role of context adjustment in refining the model's classification of attack types. Even when facing imbalanced attack distributions or weak feature signals, the model shows strong adaptability and discriminative capability. This makes it suitable for diverse and highly complex cloud-based malicious behavior detection scenarios.

4) Comparative experiment of model complexity and inference efficiency

This paper also gives a comparative experiment on model complexity and reasoning efficiency, and the experimental results are shown in Figure 5.

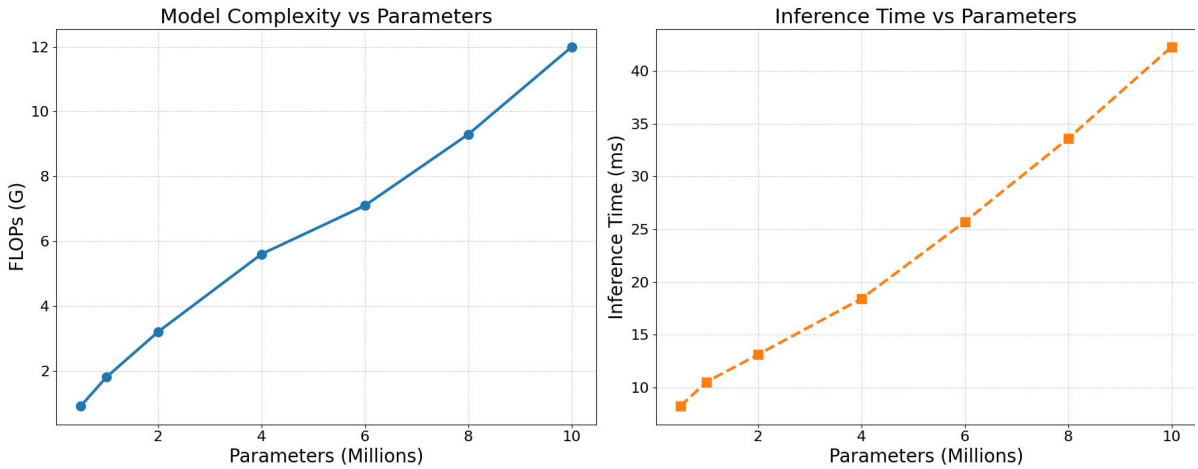


Figure 5. Comparative experiment of model complexity and inference efficiency

As shown in Figure 5, both computational complexity (FLOPs) and inference time increase steadily as the model size grows. This trend indicates that expanding the model enhances its expressive power but also leads to linear or even super-linear growth in resource consumption. In particular, when the number of parameters exceeds 6 million, the increase in inference time accelerates. This suggests that large models demand significantly more computational resources for practical deployment.

The left chart shows how model complexity changes with the number of parameters. FLOPs increase from 0.9 billion to 12.0 billion. This reflects the substantial growth in computational paths after integrating structural enhancement modules such as Multi-Scale Neighbor Attention (MSNA) and Context-Aware Attention Adjustment (CAAA). The results indicate that the improved model is more complex but also more

expressive and discriminative. It can better capture multi-level relationships and semantic features in user behavior graphs, supporting more accurate malicious user detection.

The right chart further confirms the cost of structural enhancements from the perspective of inference efficiency. As the model size increases, the inference time rises from 8.2 milliseconds to 42.3 milliseconds. Although this remains within an acceptable range, it highlights the importance of inference speed in real-world deployments, especially on cloud platforms where response latency is critical. This underscores the need to balance structural complexity and real-time performance in cloud-based applications. Overall, the experimental results confirm that the proposed improvements enhance detection performance while increasing computational load. However, this growth is meaningful and remains manageable under current hardware conditions. In the future, techniques such as model pruning, distillation, or structural compression may be applied to optimize inference efficiency while preserving detection capability. This will help meet the demands of large-scale concurrent detection in cloud security environments.

5) Stability test of the model under the condition of varying node density

This paper also presents a stability test of the model under the condition of varying node density, aiming to investigate how changes in the structural density of the behavior graph influence the model's ability to maintain consistent recognition performance. In this evaluation, the node density is systematically adjusted to simulate different real-world scenarios ranging from sparse to highly connected user behavior graphs. Such variation in density directly affects the number and distribution of neighbor relationships that each node maintains, which in turn impacts the information propagation and feature aggregation within the graph neural network. The purpose of this test is to assess whether the proposed model can remain robust and effective when exposed to graphs with different topological characteristics, as might occur in dynamic or heterogeneous cloud environments. Throughout the testing process, other variables are kept constant to ensure that observed effects are solely attributable to changes in node density. This allows for a focused examination of the model ' s structural sensitivity and its resilience to fluctuating graph connectivity patterns. The experimental setup and corresponding outcomes of this stability analysis are clearly illustrated in Figure 6.

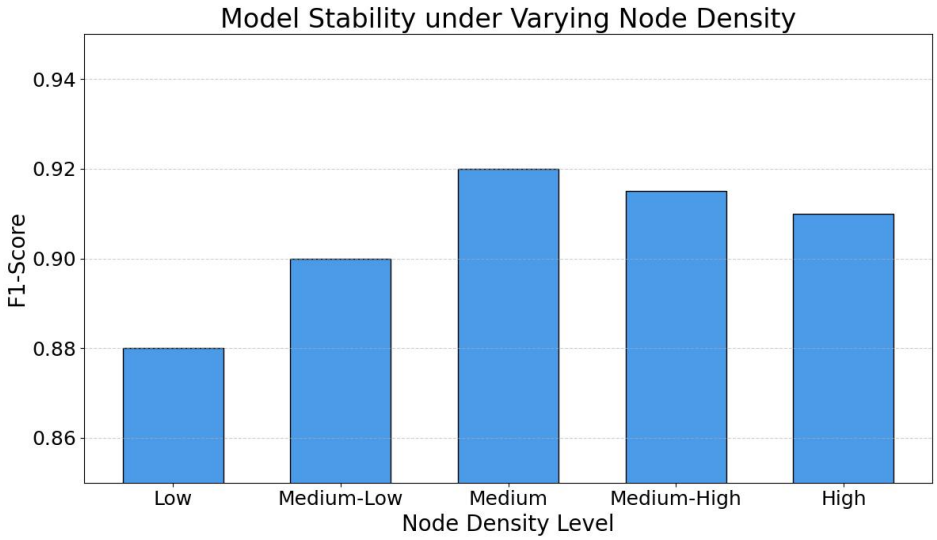


Figure 6. Stability test of the model under the condition of varying node density

As shown in Figure 6, the model maintains high stability under different node density conditions. The fluctuation range of the F1-Score is small and remains above 0.88 throughout. As node density increases from “Low” to “Medium,” the model's performance improves. The F1-Score rises from 0.88 to 0.92. This

indicates that moderately increasing the connections between nodes helps the model capture more complete neighbor information, which enhances the accuracy of behavior classification.

When the node density reaches the “Medium-High” and “High” levels, the performance slightly declines but still stays within a high-performance range. This suggests that while denser graph structures provide more information, they may also introduce redundant or noisy connections. These can reduce the attention mechanism's ability to focus on important neighbors. This result is related to the proposed Multi-Scale Attention mechanism. Although it offers some capacity for feature compression in dense graphs, it can still face challenges of information overload, leading to less precise classification.

In addition, the model shows significant improvement in robustness as the graph moves from sparse to moderate density. This validates the ability of the Context-Aware Attention Adjustment (CAAA) mechanism to preserve semantic relevance under different graph scales. It also indicates that in real cloud environments, where user behavior graphs can vary greatly in structure, the model can still adapt well and produce stable results. This reflects strong transferability and generalization across scenarios. In summary, these experimental results further confirm the practicality and robustness of the proposed method from a structural perspective. Whether in sparse graphs with fewer nodes or in dense graphs with frequent interactions, the model consistently maintains stable classification performance. This demonstrates its practical potential as a cloud-based security detection framework.

5. Conclusion

This study addresses the problem of malicious user behavior detection in cloud computing environments. It proposes an intelligent classification method based on an Improved Graph Attention Network (IGAT). The method integrates a Multi-Scale Neighbor Attention mechanism and a Context-Aware Adjustment strategy. These improvements enhance the model's expressive power and classification accuracy on complex graph structures. By modeling multi-level neighbor relationships in user behavior graphs and dynamically adjusting attention weights using historical semantic information, the proposed method demonstrates strong capability in detecting abnormal behaviors. It shows particularly high accuracy and robustness in identifying concealed and complex attack patterns. Experimental results show that the proposed method outperforms existing graph neural network models across multiple mainstream performance metrics. It maintains stable performance under different attack types, node densities, and model complexities. In tasks involving complex structures and blurred behavior boundaries, the dual mechanisms compensate for the limitations of traditional GAT in single-scale modeling and static attention allocation. The method offers both theoretical value and technical innovation. It also provides a practical foundation for deployment in large-scale cloud platforms, showing strong potential for engineering applications.

From an application perspective, this research may have a positive impact on fields such as cloud computing, IoT security, and edge intelligence. As critical information infrastructure, cloud platforms are increasingly targeted by malicious actors. Intelligent models with graph-based structural modeling and semantic analysis capabilities will become important components of next-generation security defenses. Graph neural network-driven detection systems can enable more fine-grained user behavior modeling and more dynamic attack tracking, supporting the construction of intelligent and adaptive security frameworks. Future research can be expanded in several directions. One direction is to introduce dynamic graph neural networks, reinforcement learning, or federated learning to address the challenge of attack evolution across time and platforms. Another direction is to optimize the model's inference efficiency in high-density graphs by using techniques such as model compression and knowledge distillation. This will improve performance in edge computing or resource-constrained environments. Additionally, applying the model to other domains such as financial fraud detection or anomaly detection in social networks will help validate its generalizability and scalability, supporting intelligent security across multiple fields.

References

[1] Arunkumar M, Ashok Kumar K. Malicious attack detection approach in cloud computing using machine learning techniques[J]. Soft Computing, 2022, 26(23): 13097-13107.

[2] Zulifqar I, Anayat S, Khara I. A review of data security challenges and their solutions in cloud computing[J]. International Journal of Information Engineering and Electronic Business, 2021, 12(3): 30.

[3] Veeraiah D, Mohanty R, Kundu S, et al. Detection of malicious cloud bandwidth consumption in cloud computing using machine learning techniques[J]. Computational Intelligence and Neuroscience, 2022, 2022(1): 4003403.

[4] Saxena D, Singh A K. OSC-MC: Online secure communication model for cloud environment[J]. IEEE Communications Letters, 2021, 25(9): 2844-2848.

[5] Kaja D V S, Fatima Y, Mailewa A B. Data integrity attacks in cloud computing: A review of identifying and protecting techniques[J]. Journal homepage: www. ijrpr. com ISSN, 2022, 2582: 7421.

[6] Ranjan R, Kumar S S. User behaviour analysis using data analytics and machine learning to predict malicious user versus legitimate user[J]. High-confidence computing, 2022, 2(1): 100034.

[7] Khadidos A, Subbalakshmi A, Khadidos A, et al. Wireless communication based cloud network architecture using AI assisted with IoT for FinTech application[J]. Optik, 2022, 269: 169872.

[8] Veličković, P. (2023). Everything is connected: Graph neural networks. Current Opinion in Structural Biology, 79, 102538.

[9] Veličković P. Everything is connected: Graph neural networks[J]. Current Opinion in Structural Biology, 2023, 79: 102538.

[10] Wu L, Cui P, Pei J, et al. Graph neural networks: foundation, frontiers and applications[C]//Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. 2022: 4840-4841.

[11] Gao C, Zheng Y, Li N, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions[J]. ACM Transactions on Recommender Systems, 2023, 1(1): 1-51.

[12] Reiser P, Neubert M, Eberhard A, et al. Graph neural networks for materials science and chemistry[J]. Communications Materials, 2022, 3(1): 93.

[13] Khemani B, Patil S, Kotecha K, et al. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions[J]. Journal of Big Data, 2024, 11(1): 18.

[14] Zhang M, Li P. Nested graph neural networks[J]. Advances in Neural Information Processing Systems, 2021, 34: 15734-15747.

[15] Wu L, Chen Y, Shen K, et al. Graph neural networks for natural language processing: A survey[J]. Foundations and Trends® in Machine Learning, 2023, 16(2): 119-328.

[16] Wang X, Zhang M. How powerful are spectral graph neural networks[C]//International conference on machine learning. PMLR, 2022: 23341-23362.

[17] Lin L, Huang Y, Xu L, et al. Better adaptive malicious users detection algorithm in human contact networks[J]. IEEE Transactions on Computers, 2022, 71(11): 2968-2981.

[18] Swinney C J, Woods J C. A review of security incidents and defence techniques relating to the malicious use of small unmanned aerial systems[J]. IEEE Aerospace and Electronic Systems Magazine, 2022, 37(5): 14-28.

[19] Blauth T F, Gstrein O J, Zwitter A. Artificial intelligence crime: An overview of malicious use and abuse of AI[J]. Ieee Access, 2022, 10: 77110-77122.

[20]He X, Gong Q, Chen Y, et al. DatingSec: Detecting malicious accounts in dating apps using a content-based attention network[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(5): 2193-2208.

[21]Shah H, Shah D, Jadav N K, et al. Deep learning-based malicious smart contract and intrusion detection system for IoT environment[J]. Mathematics, 2023, 11(2): 418.

[22]Brinda V, Bhuvaneshwari M. Identifying malicious secondary user presence within primary user range in cognitive radio networks[J]. Wireless Personal Communications, 2022, 122(3): 2687-2699.

[23]Kumi S, Lim C H, Lee S G. Malicious URL detection based on associative classification[J]. Entropy, 2021, 23(2): 182.

[24]Gupta R, Patel M M, Shukla A, et al. Deep learning-based malicious smart contract detection scheme for internet of things environment[J]. Computers & Electrical Engineering, 2022, 97: 107583.

[25]Wang Z, Ren X, Li S, et al. A malicious URL detection model based on convolutional neural network[J]. Security and Communication Networks, 2021, 2021(1): 5518528.

[26]Lin W, Xia C, Wang T, et al. Input and output matter: malicious traffic detection with explainability[J]. IEEE Network, 2024.

[27]Sharma R, Almáši M, Nehra S P, et al. Photocatalytic hydrogen production using graphitic carbon nitride (GCN): A precise review[J]. Renewable and Sustainable Energy Reviews, 2022, 168: 112776.

[28]Zhao J, Yan Z, Zhou Z Z, et al. A ship trajectory prediction method based on GAT and LSTM[J]. Ocean engineering, 2023, 289: 116159.

[29]Liu J, Ong G P, Chen X. GraphSAGE-based traffic speed forecasting for segment network with sparse data[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(3): 1755-1766.

[30]Wang H, Leskovec J. Unifying graph convolutional neural networks and label propagation[J]. arXiv preprint arXiv:2002.06755, 2020.