ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

False Alarm Suppression for Temporal Graph Neural Networks in Computer Network Intrusion Detection

Yafei Li[1,*], *Yan Liu*[1], *Jinpeng Chen*[1] *and Lu Zhong*[1]

[1] Department of Information Engineering, School of Rail Transit, Southwest Jiaotong University Hope College, Chengdu 610400, Sichuan, China

SUMMARY: *With the expansion of network services and the continuous complexity of attack forms, intrusion detection systems face problems such as high false positive rate and difficult to distinguish boundary samples in high dynamic traffic scenarios. Focusing on the demand of false positive suppression in computer network intrusion detection, this paper proposes a temporal graph neural network model for complex traffic scenarios. The method establishes the network traffic temporal correlation graph by session reconstruction and time window division, and integrates topology dependence extraction, behavior evolution modeling, re-discrimination of easily confused samples and false alarm constrained loss optimization to enhance the ability to identify the difference between normal fluctuations and real attacks. The experimental results based on NSL-KDD, UNSW-NB15 and CIC-IDS2017 datasets show that the Accuracy of the model reaches 96.1%, the F1-score is 95.4%, the AUC is 0.986, the FAR is reduced to 3.2%, and the Specificity is improved to 96.6%. In the NSL-KDD fine-grained attack type analysis, it also maintains a good detection effect on R2L and U2R covert attacks. The research shows that this method can effectively compress the false positive propagation space while ensuring the detection accuracy, which provides a feasible path for intelligent intrusion detection in complex network environment.*

KEYWORDS: *Network intrusion detection; Temporal graph neural network; False alarm suppression; Abnormal traffic analysis*

1 Introduction

With the continuous popularization of cloud computing, Internet of things, mobile terminals and industrial Internet, the connection scale, business types and data exchange frequency of computer networks are increasing, and the openness and complexity of cyberspace are also enhanced. Large-scale heterogeneous traffic not only improves the efficiency of information transmission, but also makes network attacks show the characteristics of concealment, continuous and compound. Behaviors such as port scanning, denial of service, botnet control, privilege escalation, and masquerading communication often no longer appear in the form of isolated events, but are embedded in the long, multi-stage interaction process, which brings higher difficulties to security monitoring. We believe that intrusion detection system, as a key part of network security protection system, assumes the important tasks of identifying abnormal access, discovering potential threats and assisting security decision-making. The accuracy of intrusion detection system directly affects the stability of network operation and the timeliness of protection response.

*wsbebk@126.com
https://doi.org/10.65102/is2026550
Published: 30 April, 2026

From the existing application situation, the high false positive rate is always an unavoidable practical problem in the field of network intrusion detection. Some normal traffic has local similarities with attack behavior in transmission mode, connection frequency, session duration and other aspects, which are easy to be misjudged by the detection model as abnormal events. In complex business scenarios, normal behaviors such as burst access, batch requests, and cross-node cooperative communication may also trigger abnormal alarms. We find that such false positives not only increase the analysis burden of security personnel and crowd out alarm disposal resources, but also may weaken the system's ability to focus on real attacks and reduce the overall defense efficiency. Especially in the high throughput and dynamic network environment, if the model lacks a comprehensive understanding of traffic context, node association and behavior evolution process, the detection results will often stay at the level of fragmentation judgment, and it is difficult to balance sensitivity and stability.

Traditional intrusion detection methods mostly rely on rule matching, feature engineering or static classification framework, which have fast response ability to known attacks. However, their adaptability is obviously insufficient when facing deformation attacks, weak feature attacks and cross-time penetration behaviors. In recent years, deep learning methods have shown strong representation learning ability in traffic identification and anomaly detection, which provides a new technical path for complex network behavior modeling. Graph neural network can describe the correlation structure among host, session, connection and communication topology, which is suitable for mining spatial dependence in network behavior. The time series modeling method is helpful to describe the evolution law of attack activities in different time slices, and improve the recognition ability of persistent abnormal patterns. Combining the two methods, the collaborative modeling of network traffic structure characteristics and time dynamic characteristics can be realized at a higher level. However, we also note that only spatio-temporal feature extraction is not enough to naturally solve the problem of false positives, and false positives may still persist for a long time without special constraints for confusing samples, boundary samples, and normal abnormal overlap regions.

Based on this, this paper focuses on the requirements of false positive suppression in computer network intrusion detection, and constructs a temporal graph neural network detection model for complex traffic scenarios. Through the network traffic temporal correlation graph representation, topology dependence and behavior evolution joint feature extraction, false positive suppression discrimination mechanism and output layer constraint optimization. Improve the ability of the model to identify the difference between real attacks and normal fluctuations. We hope to further reduce the level of false positives while ensuring the detection performance, so as to provide more practical technical support for intelligent intrusion detection in high complexity network environments.

2 Review of related research

2.1 Research status of false positive problem in network intrusion detection

The false positive problem in network intrusion detection is essentially the discrimination bias caused by the local overlap between normal traffic fluctuation and abnormal attack behavior in the feature space. Moore et al. pointed out in their study that intrusion detection systems produce a large number of alarms, and a high proportion of them may be false alarms, which not only increases the cost of manual judgment, but also weakens the actual usability of the system. Based on this understanding, this study proposed a hierarchical deep learning intrusion detection framework, which compressed the spread range of false positives by

screening suspicious traffic in stages, and finally reduced the incidence of false positives by 87.52%, indicating that hierarchically splitting the detection process is an effective idea to alleviate the accumulation of false positives [1].

On this basis, Talpini et al. further shifted the focus of research from "improving classification scores" to "improving decision reliability". This paper argues that many machine learning detectors often give too high confidence output in the face of misclassified samples or unknown class attacks, which makes it difficult to distinguish between false positives and false positives in time. In order to solve this problem, the authors introduce the idea of uncertainty quantification and open set recognition, and construct a model based on Bayesian neural network to enhance the recognition ability of unknown inputs, out-of-distribution samples, and high-risk judgment results [2]. This direction shows that false alarm control is no longer just a problem of threshold regulation, but gradually evolves into a systematic improvement of the confidence expression ability of the model.

Sivamohan et al. proposed a BiLSTM-XAI framework combining feature optimization and interpretable mechanism from the perspective of Industry 4.0 scenarios. After data cleaning and normalization, this method introduces Krill herd optimization to screen key features, combines SHAP and LIME to improve the interpretation of detection results, and achieves 98.2% classification accuracy in the experiment [3]. This kind of research shows that for complex industrial network environment, false alarm suppression depends not only on the ability of time series modeling, but also on the transparent expression of the discrimination basis. Nguyen Dang et al. proposed Hybrid Regressive Classification strategy for wireless networks, which emphasized the handling of class imbalance problem while taking into account classification and trend prediction, and pointed out that the framework could help reduce false positive rate in practical deployment. And improve the ability to identify potential attacks in advance [4].

Synthesizing the existing research, it can be seen that the improvement paths around the problem of false positives have gradually expanded from simply improving detection accuracy to multiple directions such as hierarchical detection, uncertainty modeling, interpretable analysis and unbalanced learning. However, most of the methods still focus more on sequence discrimination or classification optimization, and the mining of the association topology between network communication entities, the behavior evolution across time Windows, and the structural differences of confusing samples is still insufficient, which also provides further research space for the subsequent introduction of temporal graph neural network to carry out false alarm suppression.

2.2 Research progress on time series feature modeling and graph neural network detection methods

Focusing on the dynamic change process of network traffic, temporal feature modeling has become an important direction of intrusion detection research. Abdel-Basset et al. proposed a semi-supervised spatio-temporal deep learning method to jointly learn the temporal evolution and spatial correlation of traffic through multi-scale residual temporal convolution and traffic attention mechanism in IoT networks, indicating that it is difficult to support complex attack recognition by only relying on static features [5]. Lopes et al. further regard intrusion detection as a time series classification task and design a variety of models based on time series convolution. The results show that such convolutional structures that can directly deal with time series data show strong advantages in detection effect and efficiency, and also promote the research trend of migration from recurrent networks to lightweight time series modeling [6].

On this basis, graph neural networks have been used to characterize the structural relationship between hosts, ports, sessions and traffic. Bilot et al. pointed out in their review that network flow graph and provenance graph have become the most common graph representation forms in intrusion detection, GNN can extract attack patterns from graph structure semantics, but the time dimension is still the part that needs to be strengthened in subsequent research [7]. Wang et al. proposed N-STGAT, which incorporated node status, traffic similarity and timing relationship into graph attention modeling, and achieved high detection accuracy in specific scenarios [8]. Yang et al. proposed HRNN to extract spatio-temporal semantics by combining high-order relations of hypergraph with recurrent network, which enhanced the traffic representation ability [9]. Deng et al. further emphasized the importance of edge features and multi-hop attention aggregation for intrusion detection, indicating that graph structure modeling is shifting from pure node representation to more fine-grained relational learning [10]. In general, the existing research has provided a foundation for the modeling of timing graphs, but the specific constraints for false alarm suppression are still insufficient.

3 False positive suppression model of temporal graph neural network for computer network intrusion detection

In the complex network environment, the concealment of attack behavior is enhanced, the fluctuation of normal traffic is aggravated, and the redundant accumulation of alarm information is faced with practical problems. It is difficult to stably distinguish the boundary between real intrusion and service disturbance by the detection methods relying solely on static feature matching or single timing discrimination. Especially in the scenarios of high concurrent communication, short-time bursty access and cross-node cooperative behavior, there is often a certain similarity between normal connections and attack traffic in local characteristics, which is easy to cause the model to misidentify non-malicious behavior as abnormal events. Based on this, this paper focuses on the core goal of false alarm suppression, we construct a temporal graph neural network model for computer network intrusion detection. Starting from the original network traffic, the model completed data cleaning, session reconstruction and time window division, and then established a temporal association graph according to the interaction relationship and behavior continuity of communication entities. Then, the graph neural network is used to mine the topological dependence between nodes, and the dynamic evolution process of attack behavior is characterized by combining the time series modeling mechanism. On this basis, the false alarm suppression discrimination module is introduced to perform secondary identification and boundary correction of confusing samples. Figure 1 shows the overall framework of the false alarm suppression model for temporal graph neural network.

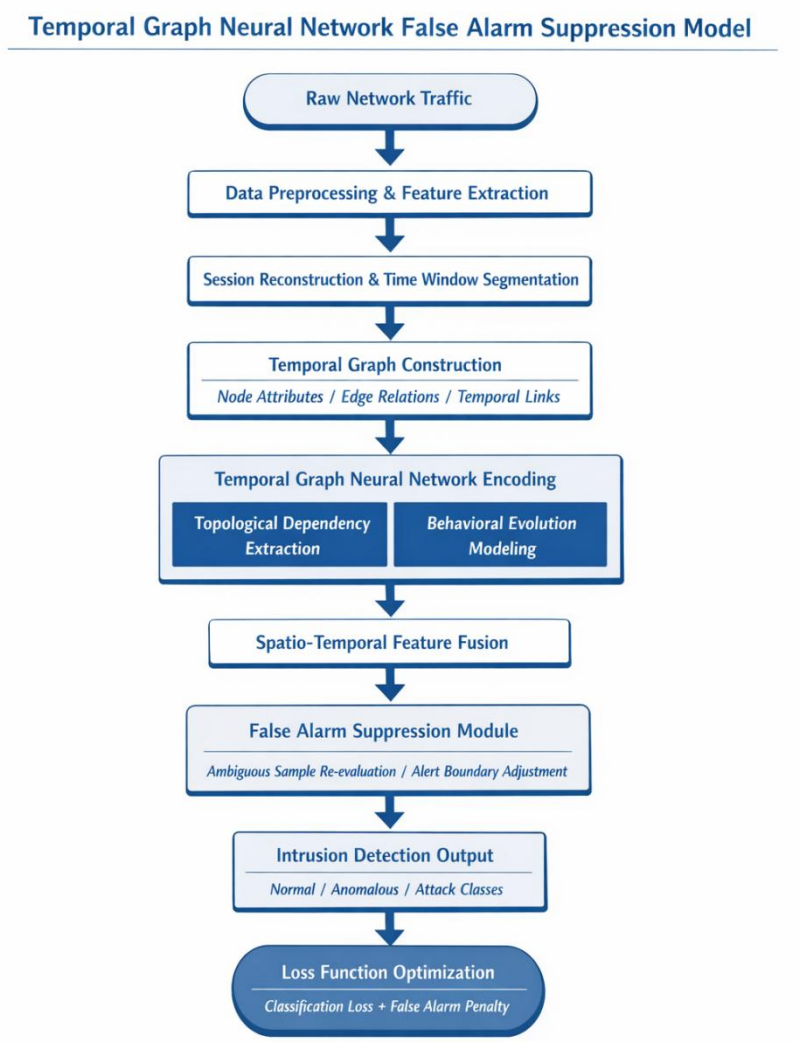

Figure 1: General framework of the neural network false alarm suppression model for temporal graph

3.1 Node representation and edge relationship modeling of network traffic temporal correlation graph

Intrusion detection is not faced with isolated data points, but with a set of continuously changing and interrelated communication behaviors. A single port probe may leave traces of similar connections in multiple time slices, and an abnormal session is often accompanied by simultaneous changes in source address, destination port, protocol field, and interaction frequency. If each flow record is still regarded as an independent sample, the model only sees a local slice, and the correlation, host cooperation and behavior continuity are difficult to be completely preserved. Based on this point, we organize network traffic as an association graph that is constantly updated with time, and express communication entities, connection behaviors and time relationships in the same structure, which provides an input basis for subsequent temporal graph neural networks to extract topological dependencies and dynamic patterns.

The network traffic graph obtained in time window t is as follows.

$$G_t = (V_t, E_t, X_t, A_t) \tag{1}$$

where $V_t = \{v_1, v_2, \ldots, v_N\}$ denotes the set of nodes, Et denotes the set of edges, $X_t \in \mathbb{R}^{N \times d}$ denotes the node feature matrix, and $A_t \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix. In this paper, the reconstructed flow-level session within a time window is used as the basic node unit, and each node corresponds to a segment of communication behavior with a clear start and end relationship. Key information such as source address, destination address, protocol type, port state, connection duration and packet size are reserved in each node, so that a single node itself has a more complete security semantics, rather than only a simple statistical value.

For any node vi, its original attribute vector is denoted as follows.

$$x_i^t = [b_i^t, p_i^t, d_i^t, r_i^t, f_i^t, m_i^t] \tag{2}$$

where b_i^t is the number of bytes, p_i^t is the number of packets, d_i^t is the connection duration, r_i^t is the packet arrival rate, f_i^t is the flag combination feature, m_i^t is the semantic description of the protocol and port mapping. Considering the large dimension difference between different dimensions, the original features are first standardized and then mapped to a unified representation space. The initial node embedding is written as follows.

$$h_i^{(0,t)} = W_n x_i^t + b_n \tag{3}$$

where W_n and b_n are trainable parameters. Through this mapping, traffic intensity, connection state and protocol semantics can be aligned in the same vector, which facilitates the subsequent model to identify the subtle differences between high-frequency normal access and abnormal detection behavior.

The establishment of edge relationship is not limited to the single condition of "communication or not", but to judge the association strength between nodes from three aspects: communication connection, behavior similarity and context continuity. A structural edge is established between two nodes in the same time window whenever they have a direct communication relationship, share a critical port, have close access patterns, or exhibit strong correlation in the session context. Edge weight is defined as follows.

$$a_{ij}^t = \alpha c_{ij}^t + \beta s_{ij}^t + \gamma u_{ij}^t \tag{4}$$

Here, c_{ij}^t represents the communication strength, s_{ij}^t represents the behavioral feature similarity, and u_{ij}^t represents the context association degree, and satisfies $\alpha+\beta+\gamma=1$. This modeling method can preserve both the explicit connection relationship and the implicit similarity relationship in the network, which is helpful to describe the complex attack patterns such as botnet linkage, lateral movement and continuous scanning.

Only the internal relationship of the window is not enough to reflect the evolution trajectory of the attack, so it is necessary to introduce time edges between adjacent time Windows to connect the continuous behaviors of the same entity or the continuation segments of the same session. Further introduce the time edge:

$$e_i^{t-1,t} = \exp\left(-\frac{|\Delta\tau_i|}{\sigma}\right) \tag{5}$$

Here, $\Delta\tau_i$ represents the time interval of nodes in adjacent Windows, and σ is the time decay coefficient. The closer the time distance is, the larger the edge weight is, indicating that the recent behavior is more reference significance for the current discrimination. After the time interval is extended, the historical influence is gradually weakened, which can avoid too

strong interference of premature information on the current state.

After obtaining the nodes and edges, the normalized adjacency matrix is constructed as follows.

$$\widetilde{A}_t = D_t^{-\frac{1}{2}}(A_t + I)D_t^{-\frac{1}{2}} \tag{6}$$

Here, D_t is the degree matrix and I is the identity matrix. After adding self-connection, nodes can retain their own characteristics when aggregating neighborhood information, and normalization operation helps to stabilize the information propagation strength between nodes with different degrees. After the above modeling, the original flow records with discrete distribution are organized into a temporal correlation graph with structural relationships and time constraints. The subsequent model can not only focus on the abnormal connection at a certain time, but also track the behavior change process along the time axis, thus providing a more solid structural foundation for false alarm suppression. Figure 2 shows the schematic diagram of network traffic timing association graph construction.

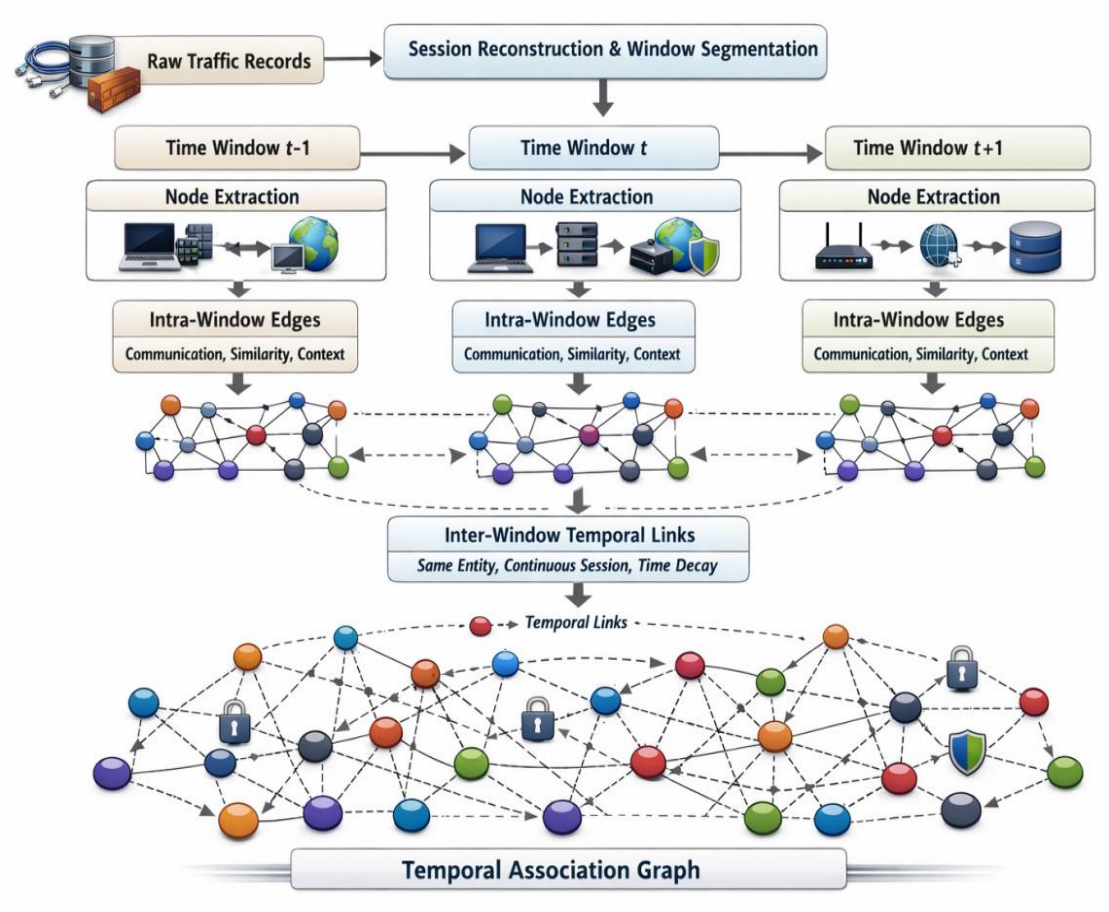

Figure 2: Schematic diagram of network traffic timing correlation graph construction

3.2 Feature extraction mechanism of temporal graph combining topology dependence and behavior evolution

After constructing the network traffic temporal correlation graph, the focus of the next step is not to simply stack the node attributes, but to extract the discriminative information that can distinguish normal communication from abnormal intrusion. Network attacks usually do not

only leave traces on a single node, but are more likely to show a composite process of multi-entity linkage, local topology anomaly aggregation, and continuous evolution across time slices. The scanning behavior will form high-frequency adjacent diffusion in a short time, the lateral movement will be accompanied by the gradual extension of the access path, and the continuous penetration often shows hidden but continuous time series fluctuations. Based on this characteristic, we simultaneously establish the topology dependence extraction branch and the behavior evolution modeling branch in the feature extraction stage, and form a unified spatio-temporal representation through the gated fusion mechanism.

In the spatial topology branch, the neighborhood aggregation representation of node vi in time window t is defined as follows.

$$z_i^t = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^t W_s h_j^{(0,t)} \tag{7}$$

Here, W_s is the spatial mapping matrix, and α_{ij}^t is the structural attention weight of node v_i to neighbor v_j. To enable the model to distinguish the importance of different adjacency relations, the attention coefficient is calculated as follows.

$$\alpha_{ij}^t = \frac{\exp(\phi_{ij}^t)}{\sum_{k \in \mathcal{N}(i)} \exp(\phi_{ik}^t)} \tag{8}$$

$$\phi_{ij}^t = \text{LeakyReLU}\left(a_s^\top \left[W_q h_i^{(0,t)} \| W_k h_j^{(0,t)} \| W_e r_{ij}^t\right]\right) \tag{9}$$

Here, a_s is a trainable vector, W_q, W_k and W_e correspond to the transformation parameters of the central node, neighbor nodes and edge features, respectively, and r_{ij}^t represents the edge relation embedding. The role of this branch is to highlight high-risk connections, abnormal aggregation paths and critical bridge nodes, so that the potential attack propagation patterns in the topology can be more clearly expressed.

The behavior evolution branch describes the node state continuously along the time axis. Considering that the communication intensity, port access habits and protocol combinations of the same entity in multiple time Windows will change dynamically, this paper uses the time attention mechanism to weighted aggregation of historical states. Let the spatial representation of node v_i in the history window τ be z_i^τ. Then its temporal correlation weight is written as follows.

$$\beta_i^{t,r} = \frac{\exp\left((W_t^q z_i^\tau)^T (W_t^k z_i^\tau)/\sqrt{dh}\right)}{\sum_{t=t-L}^{t} \exp\left((W_t^q z_i^\tau)^T (W_t^k z_i^\tau)/\sqrt{dh}\right)} \tag{10}$$

Here, L represents the history backtracking length and dh is the hidden dimension. Based on this, the behavior evolution of the node at the current time is expressed as follows.

$$u_i^t = \sum_{\tau=t-L}^{t} \beta_i^{t,\tau} W_t^v z_i^\tau \tag{11}$$

The process is able to relate recent consecutive anomalies to long-term behavioral baselines, enabling the model to focus not only on abrupt changes at one moment in time, but also to identify suspicious trends that accumulate slowly.

The representations obtained by the spatial topology branch and the behavior evolution branch correspond to two types of information: "who is associated with at the moment" and "how has it changed in the past". In order to avoid the redundancy caused by direct splicing, this paper introduces a gated fusion strategy to adaptively allocate structural features and temporal features in different scenarios. The fusion gate vector is defined as follows.

$$g_i^t = \sigma(W_g[z_i^t \| u_i^t] + b_g) \tag{12}$$

The final spatio-temporal fusion feature is expressed as follows.

$$h_i^t = g_i^t \odot z_i^t + (1 - g_i^t) \odot u_i^t \tag{13}$$

where $\sigma(\cdot)$ is the Sigmoid function, and $\odot$ is element-wise multiplication. When the gate value is large, the model depends more on the abnormal connections in the current topological neighborhood. When the threshold value is small, the model will refer to the historical behavior evolution results more. Through this dynamic fusion method, short-term burst access and persistent penetration are no longer processed at the same scale, and the representation boundary of confusing samples will be clearer, which provides a more stable input for the subsequent false alarm suppression discrimination module. Figure 3 shows the flowchart of feature extraction by fusing topology dependence and behavior evolution.

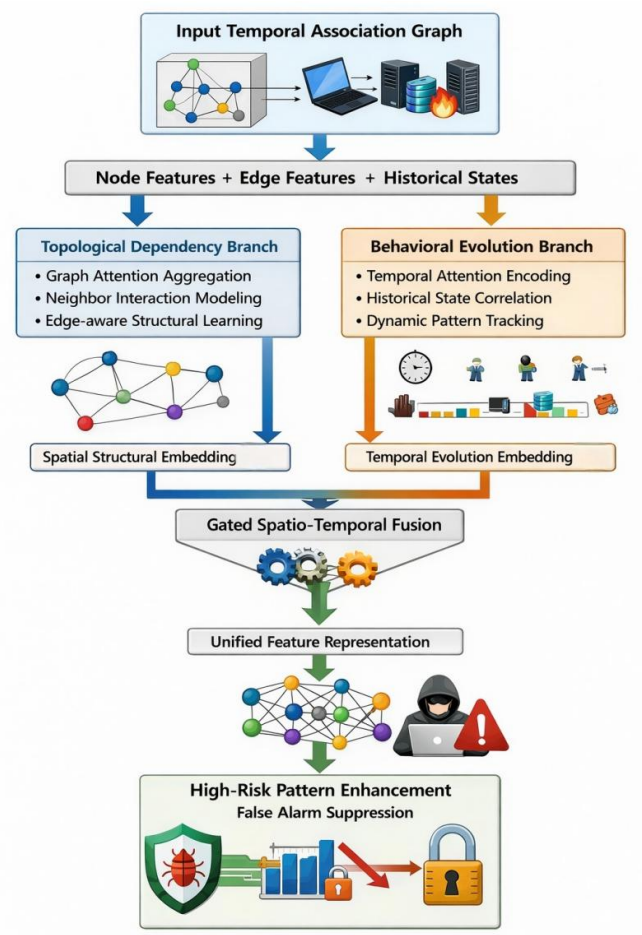

Figure 3: Flowchart of feature extraction by fusing topology dependence and behavior evolution

3.3 Design of false alarm suppression and discrimination mechanism for confusing attack behaviors

In many cases, the false positive in intrusion detection is not because the model is completely "wrong", but because some traffic segments have the local characteristics of normal traffic fluctuations and attack heurism, which leads to the squeezing of the discrimination boundary. For example, short-time high-frequency access, batch interface calls, periodic probing, and weak feature scanning may overlap with real traffic peaks at the statistical level. Relying on only a single classification output, the model is easy to push such samples directly to the abnormal class, thus amplifying false positives. Based on this phenomenon, this paper sets up a false alarm suppression discrimination mechanism after spatio-temporal fusion features. Instead of treating all alarm samples equally, the samples with unclear boundaries, contradictory contexts and large fluctuations in time state are re-evaluated, thereby reducing the probability of normal traffic being misjudged as attacks.

Let the fusion of node v_i at time t be denoted as h_i^t, and the original discriminant vector output by the classifier be denoted as follows.

$$\sigma_i^t = W_c h_i^t + b_c \tag{14}$$

Here, Wc and bc are the discriminant layer parameters. According to the interval between the maximum value and the second largest value in the output vector, the class ambiguity of the sample can be defined as follows.

$$\delta_i^t = 1 - \frac{o_{i,(1)}^t - o_{i,(2)}^t}{|o_{i,(1)}^t| + |o_{i,(2)}^t| + \varepsilon} \tag{15}$$

Here, $o_{i,(1)}^t$ and $o_{i,(2)}^t$ denote the first and second largest category responses, respectively, and ε is a tiny constant that prevents the denominator from being zero. This index is used to describe the competition intensity between the main discriminant category and the secondary discriminant category of the sample. When the two types of responses are close, the sample is more likely to be in the normal and abnormal overlap region, and it needs to further combine the neighborhood context and time state for re-discrimination.

Looking only at the class interval is still not enough, because some abnormally high scoring samples are still closer to normal traffic in local structure. To this end, this paper introduces the neighborhood consistency constraint to jointly judge the context state around the alarm sample. Let $\mathcal{B}_i^t$ be the reference neighborhood of node vi in the current graph, then its normal context consistency is defined as follows.

$$\psi_i^t = \frac{1}{|\mathcal{B}_i^t|} \sum_{j \in \mathcal{B}_i^t} \omega_{ij}^t \cdot \mathbb{I}(y_j = 0) \tag{16}$$

Here, ω_{ij}^t represents the neighborhood influence weight, $I(\cdot)$ is the indicator function, and $y_j=0$ means that the neighbor node is judged to be normal. If most of the nodes around a high-risk sample remain in a normal state for a long time, the rationality of directly triggering a strong alarm for the sample will decrease.

Considering that false alarms are often accompanied by short-time jitter in the time dimension, this paper proceeds to construct the temporal stability index to measure the discriminative change of the same entity in consecutive Windows. The time fluctuation

coefficient is defined as follows.

$$\rho_i^t = \frac{1}{K}\sum_{k=1}^{K} \| p_i^{t-k+1} - p_i^{t-k} \|_2 \tag{17}$$

where ρ_i^t is the probability distribution of samples over each class, and K is the backtracking step size. If a node frequently swings between normal and abnormal in several consecutive time slices, it often means that the sample is in the boundary region, which is more suitable for entering the suppression discrimination process, rather than immediately outputting the strong attack conclusion.

On this basis, the false alarm suppression scoring function is constructed as follows.

$$s_i^t = \lambda_1 \delta_i^t + \lambda_2 \psi_i^t + \lambda_3 \rho_i^t \tag{18}$$

Here,λ_1,λ_2,λ_3 are the weight coefficients, and it satisfies $\lambda_1+\lambda_2+\lambda_3=1$. The score comprehensively reflects the fuzzy degree of category, the normal consistency degree of neighborhood and the time fluctuation intensity. For samples with obvious alarm tendency but simultaneously high s_i^t, the system does not directly classify them into the attack category, but performs suppression correction. The revised alarm intensity is written as follows.

$$\tilde{r}_i^t = r_i^t \cdot (1 - s_i^t) \tag{19}$$

Here, r_i^t is the original abnormal response value and $\tilde{r}_i^t$ is the corrected result. The final discriminant rule is set as follows.

$$\hat{y}_i^t = \begin{cases} 0, & \tilde{r}_i^t < \tau_n \\ 1, & \tilde{r}_i^t > \tau_a \\ \text{buffer}, & \tau_n \le \tilde{r}_i^t \le \tau_a \end{cases} \tag{20}$$

Here, τ_n and τ_a are the normal and attack thresholds, respectively, and the samples in the middle region enter the buffer discrimination zone, waiting for the subsequent window to supplement the evidence. Through this design, the model reexamines high-risk samples by combining spatial context and temporal trajectory, which makes false alarm suppression shift from simple threshold filtering to a more targeted boundary correction process.

3.4 Intrusion detection output optimization and false alarm constraint loss function construction

After spatio-temporal feature extraction and false alarm suppression discrimination, the model has been able to obtain a more stable sample representation, but the final detection effect still depends on the way the output layer depicts the class boundary. In the actual network environment, there is no absolutely clear boundary between normal traffic and some weak attack behaviors. Once the output layer excessively pursues high response, the model is easy to push the normal samples near the boundary to the abnormal category, resulting in the accumulation of false positives. To alleviate this problem, we introduce a joint optimization strategy of confidence calibration and false alarm constraint in the output stage, so that the detection results not only retain the ability to identify attacks, but also avoid unnecessary amplification of abnormal scores.

Let the final fused feature of node vi at time t be hit, and the category response is obtained

by the two-layer mapping first. The hidden layer representation is defined as follows.

$$q_i^t = \mathrm{ReLU}(W_o^{(1)} h_i^t + b_o^{(1)}) \tag{21}$$

We further obtain the temperature scaled class probability distribution:

$$p_{i,c}^t = \frac{\exp(z_{i,c}^t / T)}{\sum_{r=0}^{C-1} \exp(z_{i,r}^t / T)} \tag{22}$$

Here, $z_i^t = W_o^{(2)} q_i^t + b_o^{(2)}$, C denotes the total number of categories, and T is the temperature coefficient. The purpose of introducing temperature scaling is to weaken the overconfident output of the model on the boundary samples and make the probability distribution smoother, thus leaving room for adjustment of the false alarm control.

Considering that the false positive is essentially "normal samples are pushed into the attack side", we first construct the base classification loss with class-sensitive weights:

$$L_{cls} = -\frac{1}{N} \sum_{i=1}^{N} \eta_{y_i} (1 - p_{i,y_i}^t)^{\gamma} \log p_{i,y_i}^t \tag{23}$$

Here, η_{y_i} is the class weight and γ is the difficult sample adjustment coefficient. This formula can reduce the dominant effect of a large number of easy to classify samples on training, and make the model pay more attention to difficult to distinguish traffic and boundary alarm samples.

The classification loss alone is not enough to specifically suppress false positives, so we further introduce a normal sample attack bias penalty term:

$$L_{fa} = \frac{1}{N_0} \sum_{i:y_i=0} \sum_{c=1}^{C-1} \max(0,\, p_{i,c}^t - p_{i,0}^t + m) \tag{24}$$

Here, N_0 is the number of normal samples and m is the safe interval. This constraint directly acts on the normal samples, once an attack class probability exceeds the normal class probability and breaks the interval m, the loss increases, and suppressing normal traffic from the training phase is over-alarmed.

In order to make the normal class more concentrated in the representation space, we proceed to construct the normal center compact constraint as follows.

$$L_{nc} = \frac{1}{N_0} \sum_{i:y_i=0} \| h_i^t - \mu_0 \|_2^2 \tag{25}$$

Here, μ_0 is the normal class feature center. It can compress the distribution radius of normal samples, reduce the discrete diffusion caused by the fluctuation of traffic, and make the short-time burst access not easily deviate from the normal region.

At the same time, aiming at the problem that the output of the same entity oscillates frequently in consecutive Windows, a temporal smoothness constraint is added:

$$L_{ts} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\omega_k \left\| p_i^t - p_i^{t-k} \right\|_1 \tag{26}$$

Here, ω_k is the weight decreasing with time. This term does not require that the output be exactly the same at all times, but rather limits sharp fluctuations that are not supported by obvious evidence, thereby reducing false positives triggered by transient abnormal spikes.

Finally, the overall loss function is constructed as follows.

$$L = L_{cls} + \lambda_1 L_{fa} + \lambda_2 L_{nc} + \lambda_3 L_{ts} \tag{27}$$

Here,λ_1,λ_2, and λ_3 are the balance coefficients. Through this joint optimization method, the output layer no longer only pursued higher classification scores, but synchronously constrained the stability of normal class, the discrimination of attack class and the time continuity, so that the model could control the false alarm expansion more effectively while ensuring the detection sensitivity.

4 Experimental Evaluation

4.1 Experimental Design

In order to verify the effectiveness of the false alarm suppression model of temporal graph neural network constructed in this paper in a complex network environment, the experimental part focuses on three aspects: data set construction, comparison method setting and evaluation scheme design, and tries to ensure that the results can truly reflect the comprehensive performance of the model in detection accuracy and false alarm control.

(1) Experimental data set and preprocessing scheme

Three types of public network security data sets, NSL-KDD, UNSW-NB15 and CIC-IDS2017, are selected in the experiment, covering traditional intrusion behavior, modern mixed attack traffic and abnormal communication records under complex business background. Considering the differences in field structure, time granularity and class distribution between different data sets, we first align the original traffic, deal with missing values, clean outliers and unify class labels. Then we reconstruct the flow level session according to key information such as source address, destination address, port and protocol, and segment it into continuous sample sequences according to a fixed time window. On this basis, the features of byte number, packet number, connection duration, packet arrival interval, protocol state and port semantics are further extracted to construct the node attributes and edge relationships required by the timing association graph. The experimental data set and sample composition are shown in Table 1.

Table 1: Experimental data set and sample composition

Dataset	Number of Samples After Preprocessing	Number of Normal Samples	Number of Attack Samples	Number of Attack Categories	Split Ratio (Training/Validation/Test)
NSL-KDD	125973	67343	58630	4	7:1:2
UNSW-NB15	175341	93000	82341	9	7:1:2
CIC-IDS2017	282186	148520	133666	7	7:1:2

It can be seen from Table 1 that the three types of datasets have obvious differences in sample size and attack type distribution, which can better cover the adaptability verification requirements of the false alarm suppression model in multiple scenarios.

(2) Compare model and parameter Settings

In order to highlight the improvement effect of the proposed method, LSTM, BiLSTM, GCN and GAT are selected as comparison models in the experiment. Among them, LSTM and BiLSTM are used to characterize the traditional time series modeling ability, and GCN and GAT are used to reflect the graph structure learning effect. The model in this paper uses two layers of graph representation learning module and one layer of time evolution modeling module. The hidden layer dimension is set to 128, the time window length is set to 5, the batch size is set to 64, the initial learning rate is set to 0.001, the optimizer is Adam, the maximum training round is set to 100, and the early stop strategy is used to control overfitting on the validation set. Considering the joint optimization of temperature scaling and false alarm constraint in the output stage of the model, the temperature coefficient T is set to 2.0, the weight coefficients λ_1, λ_2 and λ_3 in the false alarm suppression scoring function are set to 0.40, 0.35 and 0.25, respectively, the normal threshold τn and the attack threshold τa are set to 0.35 and 0.65, respectively. In order to reduce the interference of class imbalance on false alarm statistics, the weighted sampling method is used for normal samples and attack samples in the training phase, and the key parameters are adjusted through the validation set, so that the model achieves a stable balance between detection accuracy and false alarm control.

(3) Evaluation index and experimental environment

The experiment is evaluated from two dimensions: overall detection ability and false alarm suppression ability. The overall performance was measured by Accuracy, Precision, Recall, F1-score and AUC, which reflected the overall ability of the model to distinguish normal traffic from attack traffic. The effect of False Alarm control focuses on False Alarm Rate, False Positive Rate, Specificity and false alarm rate of normal traffic, which is used to evaluate the degree of false alarm of normal business. Considering that the experiment is based on NSL-KDD, UNSW-NB15 and CIC-IDS2017, the results presented in Table 2 and Table 3 are the comprehensive statistics of the test results of the three data sets to reflect the overall performance of the model in different network scenarios. For fine-grained attack type analysis and confusion matrix visualization, the NSL-KDD test set with more representative category division is selected for display. The experimental platform is configured with Intel Xeon processor, 32 GB memory and NVIDIA RTX series GPU, and the development environment is Python, PyTorch and CUDA. By unifying the training environment and the same data division method, the influence of external factors on the comparison of results was reduced as much as possible, and a consistent evaluation basis was provided for the analysis of subsequent experimental results.

4.2 Experimental Results

(1) Comparative analysis of overall detection performance

In order to verify the comprehensive recognition ability of the proposed model in intrusion detection tasks, LSTM, BiLSTM, GCN and GAT are selected as comparison methods, and training and testing are completed on NSL-KDD, UNSW-NB15 and CIC-IDS2017 data sets respectively, and then comprehensive statistics are performed on the test results of each data set. Accuracy, Precision, Recall, F1-score, and AUC were uniformly evaluated. The overall detection performance of different models is shown in Table 2. It can be seen that the traditional time series model has a certain foundation in continuous traffic modeling, but the use of complex topology dependence is still limited. Although the graph structure model improves the representation ability of spatial relations, there is still room for

improvement in false alarm suppression and dynamic behavior identification. In contrast, the proposed model achieves optimal results on multiple indicators, indicating that the combination of temporal correlation modeling and false alarm suppression mechanism can effectively enhance the recognition ability of complex attack behaviors.

Table 2: Comparison of the overall detection performance of different models

Model	Accuracy / %	Precision / %	Recall / %	F1-score / %	AUC
LSTM	92.4	91.7	90.8	91.2	0.948
BiLSTM	93.1	92.5	91.6	92.0	0.956
GCN	94.2	93.8	93.1	93.4	0.968
GAT	94.8	94.3	93.7	94.0	0.973
Proposed	96.1	95.9	95.0	95.4	0.986

In summary, the Accuracy of the model in this paper reaches 96.1%, the F1-score reaches 95.4%, which is 1.3 and 1.4 percentage points higher than that of GAT, respectively, and the AUC increases to 0.986, indicating that the method has a more stable advantage in overall detection performance.

(2) Comparative analysis of false alarm suppression effect

The false alarm control capability is the focus of the model evaluation in this paper. To this end, the False Alarm Rate, False Positive Rate, Specificity and false alarm rate of normal traffic of each model are further statistically analyzed. The false alarm suppression effect of different models is shown in Table 3. It can be seen from the results in the table that as the model's ability to depict spatio-temporal correlation and boundary samples increases, the proportion of normal traffic misjudged as attack traffic continues to decrease. This method introduces a false alarm constraint in the output layer, and combines the re-discrimination mechanism of confusing samples mentioned above, so that the alarm results no longer rely on a single high response, but more consider the context consistency and time stability, so that the performance of normal traffic identification is more stable.

Table 3: Comparison of false alarm suppression effects of different models

Model	FAR / %	FPR / %	Specificity / %	Normal Traffic Misclassification Rate / %
LSTM	6.8	5.9	92.7	7.1
BiLSTM	6.1	5.4	93.4	6.4
GCN	5.3	4.6	94.2	5.6
GAT	4.9	4.1	94.7	5.0
Proposed	3.2	2.7	96.6	3.4

From the perspective of false alarm indicators, the FAR of the proposed model is reduced to 3.2%, 3.6 percentage points lower than that of LSTM and 1.7 percentage points lower than that of GAT, and the Specificity is improved to 96.6%, indicating that the model effectively compresses the false alarm propagation space while maintaining the detection sensitivity.

(3) Detection performance analysis under different attack types

In order to further investigate the adaptability of the model in fine-grained attack scenarios, this paper selects the NSL-KDD test set with relatively typical attack category division, makes statistics on F1-score under different attack types, and compares it with GCN and GAT. The line chart of the detection effect of different attack types is shown in FIG. 4. The results show that GCN and GAT have better performance in attack categories with obvious structural characteristics such as DoS and Probe, but there are still certain recognition

fluctuations in attack types with stronger conceit and more confusing sample boundaries such as R2L and U2R. Since the proposed model utilizes both topology dependence and behavior evolution information, and introduces false alarm suppression and boundary correction mechanism in the output stage, the recognition of weak feature attacks is more stable. Especially on U2R and R2L samples, the F1-score of the proposed model reaches 91.4%and 93.1%respectively, which are significantly higher than those of the comparison models, indicating that the proposed method has a stronger ability to distinguish confusable attack types, and further verifies the effectiveness of the false alarm suppression mechanism in the discrimination of complex boundary samples.

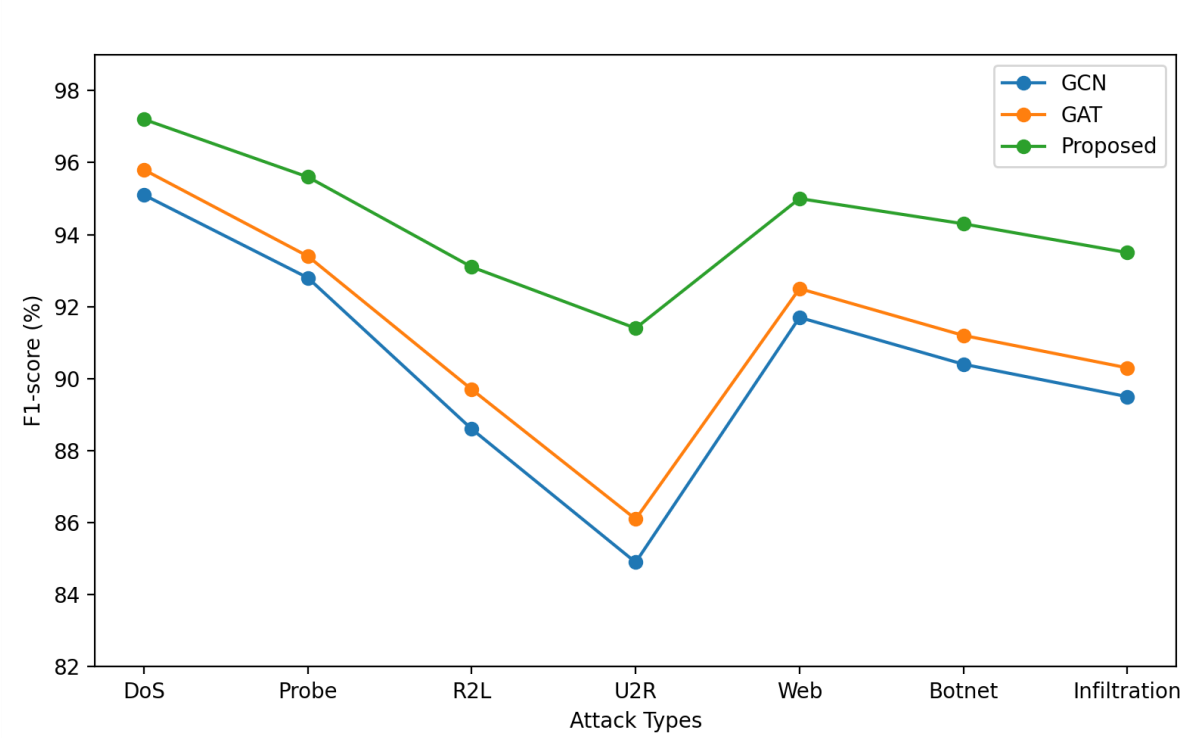

Figure 4: Line chart of detection effect of different attack types

(4) Model convergence and training stability analysis

The convergence rate and validation stability during model training are directly related to the deployability of the method. The model training convergence curve is shown in Figure 5. It can be seen that the training loss decreases rapidly in the first 15 epochs and then gradually platelows off. Although the validation loss fluctuates slightly in the middle and early stage, the overall change trend is consistent with the training set, and there is no obvious divergence. This indicates that the proposed model still maintains good optimization stability after introducing multi-branch feature extraction and false alarm constraints. By near the 40th epoch, the training loss has dropped below 0.10, and the validation loss is stable at around 0.12, and the subsequent fluctuation is small, indicating that the model has basically completed parameter convergence and the risk of overfitting is low.

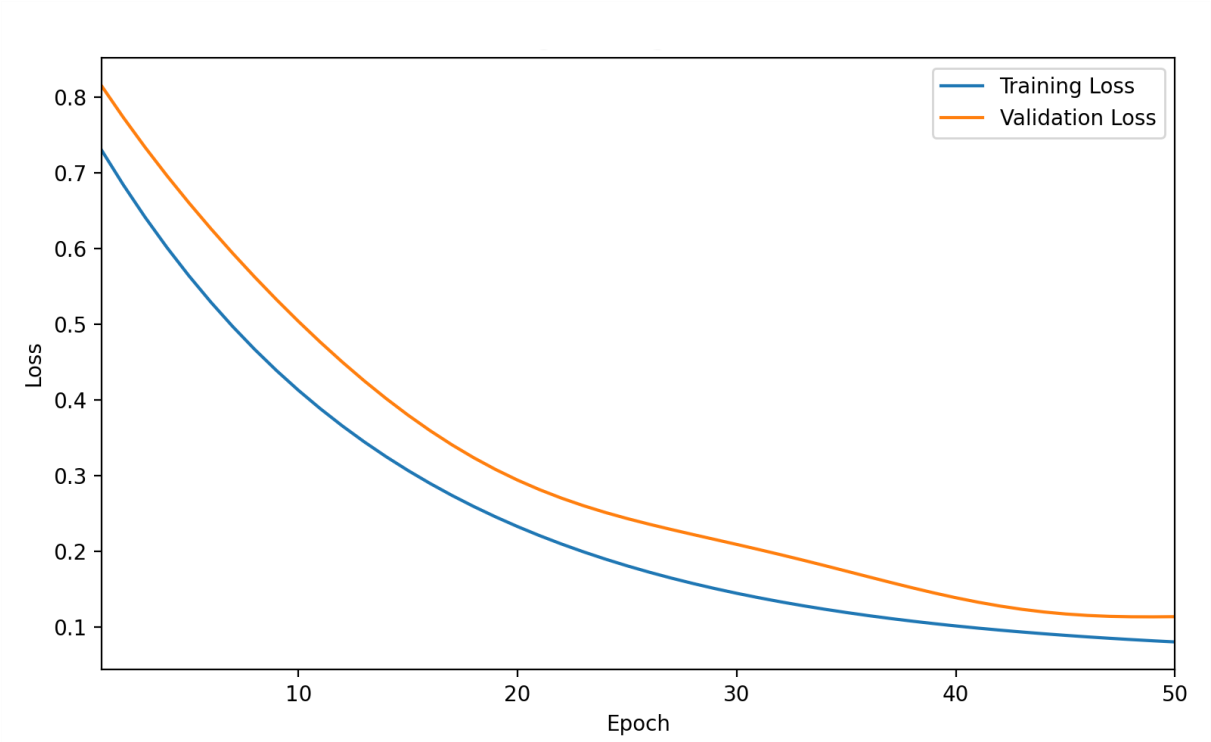

Figure 5: Convergence curve of model training

(5) Visual analysis of classification results

In order to intuitively show the discrimination effect of the proposed model on normal traffic and attack traffic, the NSL-KDD test set is further taken as an example to show the confusion matrix heat map, and the classification results are shown in Figure 6. Overall, the values in the diagonal regions are significantly higher than those in the off-diagonal regions, indicating that most samples can be correctly classified. The number of correctly identified samples in the normal class reached 2864, and only a small number of samples were misclassified into DoS, Probe and U2R categories. The recognition results of DoS and Probe categories are also more concentrated, indicating that the model can better capture the characteristics of high-frequency abnormal communication. In contrast, there is still a small amount of crossover between R2L and U2R, but the number of misclassifications is generally controllable, and no large area of confusion is formed. On the whole, the model in this paper has a clear discrimination boundary on the main categories, and especially has a good protection effect on normal traffic, which is consistent with the decline results of false alarm indicators mentioned above, and also confirms the positive effect of false alarm suppression mechanism on boundary sample correction from the perspective of visualization.

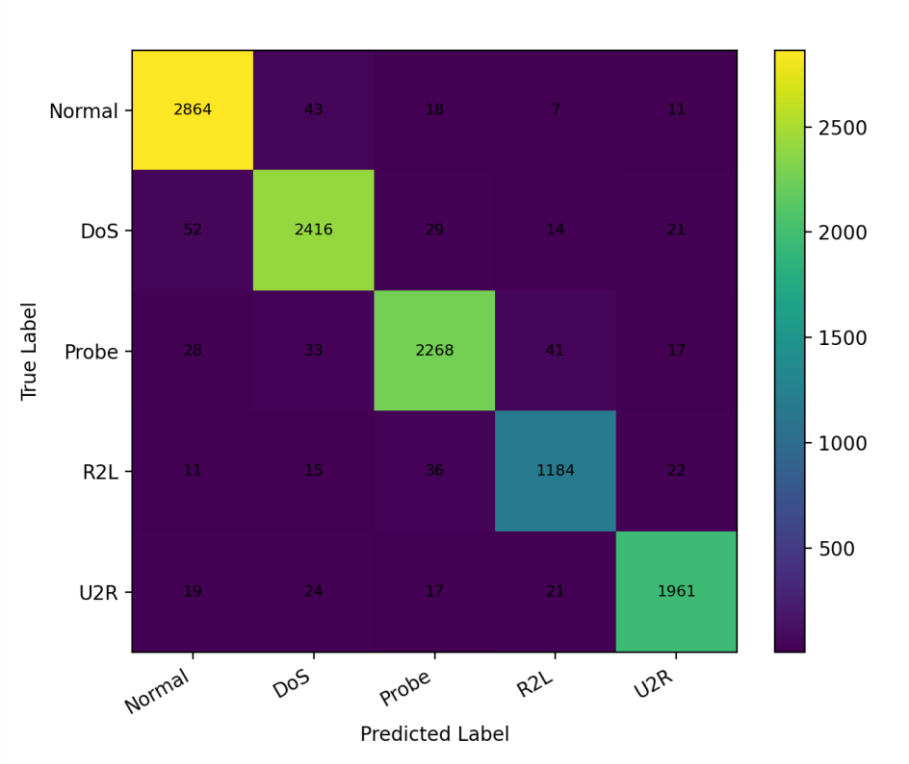

Figure 6: Heat map of the confusion matrix of the proposed model

5 Discussion

From the experimental results, the false alarm suppression model of temporal graph neural network constructed in this paper shows relatively stable advantages in overall detection performance and false alarm control ability, which indicates that it is effective to jointly incorporate topology association, behavior evolution and false alarm constraints into the same detection framework. We believe that traditional intrusion detection models tend to pay more attention to the abnormal intensity in a single flow record or a single time slice, and can quickly identify attacks with obvious characteristics. However, in the face of normal service burst, short-time high-frequency access and cross-node cooperative communication, it is easy to amplify the alarm due to local statistical anomalies. The key reason why the proposed method can achieve better results in terms of Accuracy, F1-score and FAR is that it does not regard traffic samples as independent objects, but expresses the communication relationship, context relationship and time continuity between nodes by a temporal correlation graph. Then, the fusion feature extraction mechanism is used to jointly characterize the attack diffusion path and behavior change trend, so as to enhance the identification ability of confusing samples. We also see from this that false alarm suppression depends not only on the adjustment of output layer threshold, but also on the collaborative modeling of traffic structure relationship and timing context.

The improvement of false alarm suppression effect also reflects the practical significance of the adjustment of the discriminant idea of the model in this paper. Many previous methods mainly rely on the maximum class probability to complete the decision in the output stage. Although this method is direct, it is easy to produce overconfident judgment in the boundary region. In addition to the classification results, this paper adds ambiguity evaluation, neighborhood consistency analysis and time stability constraints, so that the alarm decision is not only determined by a certain high response, but also combined with the window state and

the surrounding node environment for re-correction. In the experiment, FAR is reduced to 3.2%, and the false positive rate of normal traffic is controlled at 3.4%, which indicates that the processing method of "first identification, then suppression" can more effectively compress the false positive diffusion range. Especially in the fine-grained attack analysis of NSL-KDD test set, the model still maintains a good recognition level on the attack types with strong concealment such as R2L and U2R, which indicates that the false alarm constraint does not simply sacrifice the detection sensitivity, but improves the stability of the class boundary on the basis of retaining the attack response ability. We believe that this improvement shows that false alarm control and attack identification are not antagonistic goals, and they can be synergistically improved under a more reasonable discrimination mechanism.

However, there is still room for further improvement of the proposed method. On the one hand, temporal graph modeling and multi-branch feature extraction improve the detection accuracy, but they also bring higher training and inference overhead. In the large-scale online network environment, the number of nodes and edge relationships will increase rapidly. How to maintain the false positive suppression effect while ensuring the real-time performance of the model is still a key issue to be considered in the deployment stage. On the other hand, the experiment is mainly based on the public data set. Although the data set is representative, the service structure, protocol distribution and attack deformation mode in the real network are more complex, and the continuous adaptation ability of the model in the face of unknown attacks, hybrid attacks and concept drift needs further verification. In addition, the tolerance boundaries of false positives and false negatives are not consistent in different scenarios. How to dynamically adjust the constraint weights according to the differences in the industrial network, cloud platform or Internet of things environment is also a direction worthy of in-depth discussion in subsequent research. We also recognize that further enhancements in model lightweight, online updates, and scenario adaptation capabilities are needed to push this approach further toward real-world deployment.

In general, this study shows that the problem of false positives in intrusion detection is not only caused by unreasonable threshold setting of classifier, but also by the lack of collaborative understanding of communication structure, time evolution and boundary samples. The combination of temporal graph neural network and false alarm suppression mechanism can make up for this deficiency to a certain extent, and provide new ideas for intelligent security detection in complex network environment. Subsequent research can continue to expand in lightweight modeling, cross-domain transfer, adaptive threshold, and unknown attack recognition in open scenarios, so that the model can show stronger generalization ability and application value under conditions closer to the real business. We hope that related research can further push intrusion detection models from simply pursuing high classification scores to a comprehensive optimization path that takes into account detection accuracy, false alarm control, and scenario adaptability.

6 Conclusion

In order to solve the problems of local overlap between normal traffic and attack behavior and high false positives in traditional models in computer network intrusion detection, this paper constructs a false positive suppression model based on temporal graph neural network. Based on the network traffic temporal correlation graph, the model combines topology dependence learning and behavior evolution representation. In the detection stage, a false alarm suppression discrimination mechanism for confusing samples is introduced, and the classification stability of boundary samples is optimized through output calibration and constraint loss, so as to improve the adaptability of the model to complex traffic scenarios.

Comprehensive experimental results on NSL-KDD, UNSW-NB15 and CIC-IDS2017 datasets show that the proposed model is superior to LSTM, BiLSTM, GCN and GAT in terms of overall detection performance and false alarm control ability, with an Accuracy of 96.1%. F1-score reached 95.4%, AUC increased to 0.986, FAR decreased to 3.2%, Specificity reached 96.6%, showing good detection accuracy and false alarm suppression effect. Further fine-grained analysis on the NSL-KDD test set shows that the proposed model still maintains a relatively stable recognition ability for strong concealment attack types such as R2L and U2R, indicating that the constructed temporal correlation modeling and boundary correction mechanism have good discrimination value. The future research can be further expanded in the direction of lightweight deployment, cross-domain migration, adaptive threshold adjustment, and unknown attack recognition, so as to further improve the generalization ability and application value of the model in the real complex network environment.

References

[1] Moore S, Cruciani F, Nugent C D, Zhang S, Cleland I, Sani S. Deep learning for network intrusion: A hierarchical approach to reduce false alarms [J]. Intelligent Systems with Applications, 2023, 18: 1-13. DOI: 10.1016/j.iswa.2023.200215.

[2] Talpini J, Sartori F, Savi M. Enhancing trustworthiness in ML-based network intrusion detection with uncertainty quantification [J]. Journal of Reliable Intelligent Environments, 2024, 10: 501-520. DOI: 10.1007/s40860-024-00238-8.

[3] Sivamohan S, Sridhar S S. An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework [J]. Neural Computing and Applications, 2023, 35(15): 11459-11475. DOI: 10.1007/s00521-023-08319-0.

[4] Nguyen Dang K D, Fazio P, Voznak M. A Novel Deep Learning Framework for Intrusion Detection Systems in Wireless Network [J]. Future Internet, 2024, 16(8): 264. DOI: 10.3390/fi16080264.

[5] Abdel-Basset M, Hawash H, Chakrabortty R K, Ryan M J. Semi-Supervised Spatiotemporal Deep Learning for Intrusions Detection in IoT Networks [J]. IEEE Internet of Things Journal, 2021, 8(15): 12251-12265. DOI: 10.1109/JIOT.2021.3060878.

[6] Lopes I O, Zou D, Abdulqadder I H, Akbar S, Li Z, Ruambo F, Pereira W. Network intrusion detection based on the temporal convolutional model [J]. Computers & Security, 2023, 135: 103465. DOI: 10.1016/j.cose.2023.103465.

[7] Bilot T, El Madhoun N, Al Agha K, Zouaoui A. Graph Neural Networks for Intrusion Detection: A Survey [J]. IEEE Access, 2023, 11: 49114-49139. DOI: 10.1109/ACCESS.2023.3275789.

[8] Wang Y, Li J, Zhao W, Han Z, Zhao H, Wang L, He X. N-STGAT: Spatio-Temporal Graph Neural Network Based Network Intrusion Detection for Near-Earth Remote Sensing [J]. Remote Sensing, 2023, 15(14): 3611. DOI: 10.3390/rs15143611.

[9] Yang Z, Ma Z, Zhao W, et al. HRNN: Hypergraph Recurrent Neural Network for Network Intrusion Detection [J]. Journal of Grid Computing, 2024, 22: 52. DOI:

10.1007/s10723-024-09767-1.

[10] Deng P, Huang Y. Edge-featured multi-hop attention graph neural network for intrusion detection system [J]. Computers & Security, 2025, 148: 104132. DOI: 10.1016/j.cose. 2024.104132.

[11] Ozkan-Okay M, Samet R, Aslan O, Gupta D. A Comprehensive Systematic Literature Review on Intrusion Detection Systems [J]. IEEE Access, 2021, 9: 157727-157760. DOI: 10.1109/ACCESS.2021.3129336.

[12] Ayyagari M R, Kesswani N, Kumar M, Kumar K. Intrusion detection techniques in network environment: a systematic review [J]. Wireless Networks, 2021, 27(2): 1269-1285. DOI: 10.1007/s11276-020-02529-3.

[13] Alsoufi M A, Razak S, Siraj M M, Nafea I, Ghaleb F A, Saeed F, Nasser M. Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review [J]. Applied Sciences, 2021, 11(18): 8383. DOI: 10.3390/ app11188383.

[14] Yang Z, Liu X, Li T, Wu D, Wang J, Zhao Y, Han H. A systematic literature review of methods and datasets for anomaly-based network intrusion detection [J]. Computers & Security, 2022, 116: 102675. DOI: 10.1016/j.cose.2022.102675.

[15] Abdulganiyu O H, Ait Tchakoucht T, Saheed Y K. A systematic literature review for network intrusion detection system (IDS) [J]. International Journal of Information Security, 2023, 22: 1125-1162. DOI: 10.1007/s10207-023-00682-2.

[16] Wu C, Li W. Enhancing intrusion detection with feature selection and neural network [J]. International Journal of Intelligent Systems, 2021, 36(7): 3087-3105. DOI: 10.1002/ int.22397.

[17] Sommestad T, Holm H, Steinvall D. Variables influencing the effectiveness of signature-based network intrusion detection systems [J]. Information Security Journal: A Global Perspective, 2022, 31(6): 711-728. DOI: 10.1080/19393555.2021.1975853.

[18] Fu Y, Du Y, Cao Z, Li Q, Xiang W. A Deep Learning Model for Network Intrusion Detection with Imbalanced Data [J]. Electronics, 2022, 11(6): 898. DOI: 10.3390/ electronics11060898.

[19] Yue Y, Chen X, Han Z, Zeng X, Zhu Y. Contrastive Learning Enhanced Intrusion Detection [J]. IEEE Transactions on Network and Service Management, 2022, 19(4): 4232-4247. DOI: 10.1109/TNSM.2022.3218843.

[20] Halbouni A, Gunawan T S, Habaebi M H, Halbouni M, Kartiwi M, Ahmad R. CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System [J]. IEEE Access, 2022, 10: 99837-99849. DOI: 10.1109/ACCESS.2022.3206425.

[21] He J, Wang X, Song Y, Xiang Q. A multiscale intrusion detection system based on pyramid depthwise separable convolution neural network [J]. Neurocomputing, 2023, 530: 48-59. DOI: 10.1016/j.neucom.2023.01.072.

[22] Ren K, Yuan S, Zhang C, Shi Y, Huang Z. CANET: A hierarchical CNN-Attention model for Network Intrusion Detection [J]. Computer Communications, 2023, 205: 170-181. DOI: 10.1016/j.comcom.2023.04.018.

[23] Long Z, Yan H, Shen G, Zhang X, He H, Cheng L. A Transformer-based network intrusion detection approach for cloud security [J]. Journal of Cloud Computing, 2024, 13(1): 1-11. DOI: 10.1186/s13677-023-00574-9.

[24] Zhong M, Lin M, He Z. Dynamic multi-scale topological representation for enhancing network intrusion detection [J]. Computers & Security, 2023, 135: 103516. DOI: 10.1016/j.cose.2023.103516.

[25] Xu R, Wu G, Wang W, Gao X, He A, Zhang Z. Applying self-supervised learning to network intrusion detection for network flows with graph neural network [J]. Computer Networks, 2024, 248: 110495. DOI: 10.1016/j.comnet.2024.110495.