Temporal and Topological Enhanced Graph Neural Networks for Traffic Anomaly Detection

Minghui Gao[1,2], Zhijun Zhang[1,2], Liangliang Cui[1,2], Sibo Feng[1,2], Jingyi Liu[3] and Yongzhen Jiang[3*]

[1]*NARI Group Corporation (State Grid Electric Power Research Institute), China*
[2]*Beijing Kedong Electric Power Control System Co., Ltd., China*
[3]*Software College, Northeastern University, China*
E-mail: jiangyongzhen@stumail.neu.edu.cn
[*]*Corresponding Author*

Received 17 October 2024; Accepted 21 April 2025

Abstract

With the rapid advancement of network technologies, ensuring the security of network communications has become increasingly critical. Network traffic anomaly detection plays a pivotal role in identifying irregularities that threaten the security, reliability, and stability of cyberspace services. Recently, deep learning-based approaches, particularly those utilizing Graph Neural Networks (GNNs), have gained attention due to their powerful representation learning capabilities. However, these methods are limited by the receptive field of GNNs and their ability to capture temporal feature dependencies, leaving room for performance improvement. To address these limitations, we propose a novel GNN with a pre-characterization mechanism using PageRank to enhance the receptive field and improve accuracy without over-smoothing. Additionally, we incorporate a temporal attention module

Journal of Cyber Security and Mobility, Vol. 14_2, 457–474.
doi: 10.13052/jcsm2245-1439.1428

to capture potential temporal dependencies in the data. Our experimental results demonstrate that our method achieves a detection accuracy of 98.3%, representing a performance boost of approximately 3% compared to existing approaches.

Keywords: Anomaly detection, graph neural networks, temporal graphs.

1 Introduction

The rapid development of network technologies has made network communication capabilities a key indicator of a nation's progress. While networks have undeniably enhanced daily life's convenience and speed, they have also introduced substantial security risks. The frequent occurrence of network security incidents has resulted in considerable economic losses, underscoring the critical importance of ensuring network security.

Traffic anomaly detection is crucial for monitoring and identifying irregularities within network traffic, thereby maintaining the security, reliability, and stability of cyberspace services. The core idea behind such detection is to examine network-related parameters such as IP addresses, protocol usage, port numbers, services, and traffic measures, which are typically encoded within log files that include temporal data representing the flow of data over time within the network.

In recent years, traffic anomaly detection leveraging deep learning models [9, 18–20] has garnered significant attention due to their powerful representation learning capabilities. These models generally consist of two components: one for extracting refined representations from network-related parameters, and another for aggregating temporal information across different time frames to generate a final representation for detection purposes. The detection methods based on deep learning often format log data into matrices and utilize Convolutional Neural Networks (CNNs) to learn useful representations. However, the sliding window approach inherent to CNNs assumes that each segment of the matrix exhibits internal dependencies along both the X and Y axes, similar to the structure of an image. In these matrices, the X-axis typically represents network parameters, and the Y-axis encodes temporal information. This assumption is impractical, as the arrangement of parameters along the X-axis is often arbitrary or alphabetical, leading to weak aggregation in the X-axis.

To address this limitation, recent approaches [5, 12, 29] have try to model log files as non-Euclidean temporal graph structures, leveraging the advanced

modeling capabilities of graph data and the representation learning strengths of GNNs. However, we recognize that, at the topological level, malicious traffic does not travel directly from the attacker's device to the victim's device. Instead, it often traverses a sequence of intermediary devices to evade detection. This extended attack path necessitates an increase in the GNN's receptive field. Different from CNNs, in the context of GNNs, the receptive field is inherently linked to the depth of the model. Yet, simply increasing the depth of the model may lead to over-smoothing [2, 13], wherein the aggregated information from each node, derived from an almost complete graph, converges to a non-discriminatory state.

To overcome these challenges, we introduce a novel approach that initially processes the parameter information through a multi-layer perceptron for preliminary learning in a topology-agnostic manner. We then incorporate the PageRank algorithm to pre-characterize the devices in the entire graph. After this pre-characterization, the message-passing mechanism inherent in GNNs is employed to learn from the pre-diffused features. This hybrid approach of pre-diffusion followed by message passing effectively expands the model's receptive field. Additionally, we incorporate an attention mechanism-based module at the temporal level to identify and leverage long-range dependencies.

Additionally, we incorporate an attention mechanism-based module at the temporal level to identify and leverage long-range dependencies, which are often crucial for accurate sequence modeling.

In summary, our contributions are as follows:

- We propose a GNN-based Network Traffic Anomaly Detector that avoids the constraints of Euclidean data.
- We recognize that anomalous behavior typically involves several stages of relay and forwarding and introduce a pre-diffusion mechanism to enhance the receptive field while mitigating the issue of over-smoothing.
- Our experimental evaluations on two real-world datasets demonstrate that our method achieves an accuracy of 98.3%, showing an improvement of approximately 3% over existing methods.

2 Related Works

2.1 Classical Traffic Anomaly Detector

Statistical approaches, including Grubbs' test [27] and the ESD test [10], were initially dominant in traffic anomaly detection. These techniques operate

under the assumption that attacker behaviors significantly differ from those of legitimate users, thereby identifying deviations as anomalies.

Traditional anomaly detection methods [3, 6] have proven effective in detecting novel attack patterns. However, their performance is highly dependent on the quality of feature engineering. When feature analysis of benign traffic is insufficient, these methods tend to produce a high false-positive rate [6].

2.2 Deep Learning-based Traffic Anomaly Detector

Deep learning techniques, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have recently gained traction in traffic anomaly detection. These models utilize multiple processing layers to extract abstract representations from data. By directly deriving low-order features from raw inputs and converting them into high-order features, neural networks enable automated learning and analysis, thereby reducing the dependence on manual feature engineering [7].

Wang *et al.* [26] integrated CNNs and RNNs to enhance anomaly detection accuracy while minimizing false alarms on the DARPA1998 and ISCX2012 datasets. Farahnakian and Heikkonen [8] proposed a deep autoencoder with greedy layer-wise pre-training for intrusion detection. Similarly, Al-Qatf *et al.* [1] introduced STL-IDS, which employs sparse autoencoders for feature extraction followed by a Support Vector Machine (SVM) for intrusion detection, achieving improved classification accuracy and reduced computational time. Various studies continue to explore CNNs, LSTMs, and hybrid models for traffic anomaly detection. Despite these advancements, challenges such as low accuracy, high false alarm rates, and real-time performance constraints persist [14, 25, 26]. Shone *et al.* [22] introduced a deep learning framework utilizing stacked Nonsymmetric Deep Autoencoders (NDAEs) for unsupervised feature learning, achieving 97.85% accuracy on the KDD-Cup'99 dataset and 89.22% on the NSL-KDD dataset. CNNs and RNNs remain widely used architectures, with CNNs excelling in spatial feature extraction. For example, Vinayakumar *et al.* [24] leveraged CNNs for network intrusion detection, while Li *et al.* [17] employed CNNs for representation learning in intrusion detection tasks. Additionally, Yin *et al.* [28] proposed RNN-IDS and compared its effectiveness with traditional machine learning algorithms. LSTM, a well-known RNN variant, is particularly beneficial for processing sequential data with dependencies. Kim *et al.* [15]

demonstrated that LSTM achieves 96.93% accuracy in intrusion detection tasks.

However, (1) due to limitations in its receptive field, CNNs and RNNs struggle with capturing long-range dependencies. The receptive field size of CNNs and RNNs (or LSTMs) depends on the convolutional kernel size and memory gate stacking, respectively. The malicious traffic may not traverse directly from the attacker's device to the intended victim's device,but rather navigates through a series of intermediaries to evade detection. Attempts to expand the receptive field may lead to GPU memory constraints or gradient vanishing issues. Moreover, (2) as discussed in Section 1, CNN's sliding window mechanism assumes that matrix segments maintain internal dependencies along both the X and Y axes, akin to an image structure. In these matrices, the X-axis typically represents network parameters, while the Y-axis encodes temporal information. However, this assumption is often unrealistic, as X-axis parameter arrangements are generally arbitrary or alphabetical, resulting in weak aggregation along the X-axis.

These limitations motivate our development of a graph neural network with an extended receptive field in this paper.

2.3 Graph Neural Networks

Graph Neural Networks (GNNs) are pivotal for various tasks on graph, such as node/edge/graph level classification and link prediction. Basically, GNNs operate by passing messages between nodes along the edges of the graph, i.e., each node/edge in the graph updates its state based on the messages received from its neighbors. Then, these nodes/edges aggregate information from their neighbors to update their own state. After aggregation, an update operator is applied to the aggregated information to update the node's state so that produce the final predict. For a simple illustration, we first introduce the non-temporal graphs and their related notations. Mathematically, an unweighted graph is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ denotes the node set and the edge set of the graph. A is a symmetric adjacency matrix $A = \{0, 1\}^{|V|\times|V|}$, and A_{ij} is non-zero if there exists a edge between node i and node j. There may exist d-dimensional features for each **edge** in this graph, and these features can be described by a feature matrix $X \in \mathbb{R}^{|E|\times d}$. In this scenario, graph can also be denote as $\mathcal{G} = (A, X)$.

In this paper, we mainly consider the **edge** classification task in the graphs, since we model the traffics as edges. For such the task, we evaluate

the cross-entropy loss function over all training examples:

$$\mathcal{L}_{tra} = -\sum_{i=1}^{|\mathcal{E}_L|}\sum_{k=1}^{|F|} Y_{ik} \ln(Y'_{ik}(\mathcal{G})) \tag{1}$$

where $\mathcal{E}_L$ is the set of edge with labels, $|F|$ denotes the number of categories, Y is the real label matrix with $Y_{ik} = 1$ if node e_i belongs to category k and $Y_{ik} = 0$ otherwise, and $Y'(\mathcal{G})$ is the logits of the model.

3 Proposed Methods

3.1 Overview

As previously mentioned, anomalous traffic detection models based on temporal information typically consist of two components: one for cross-temporal aggregation and another for topology aggregation. An overview of the entire model is depicted in Figure 1.

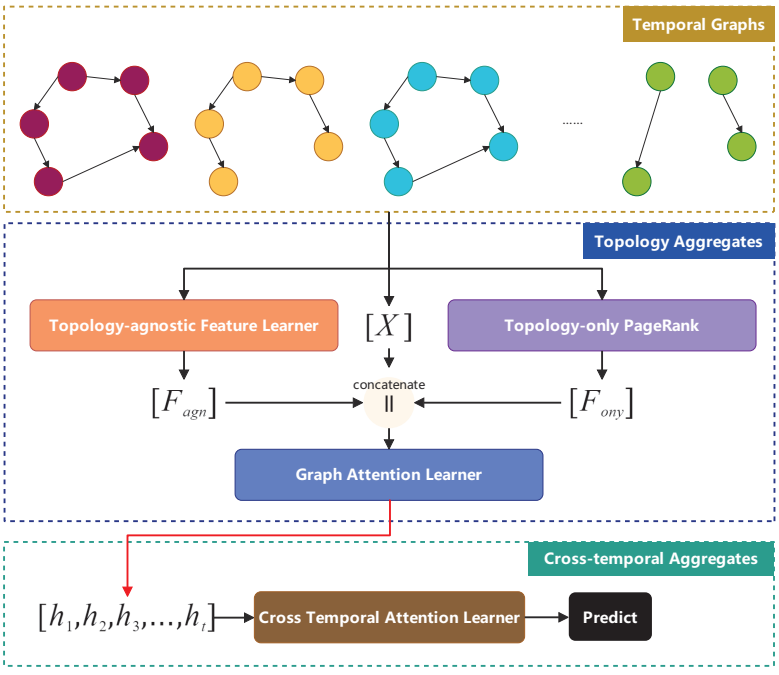

Figure 1 Overview of the proposed model.

In the topology aggregation module, we first process the original features X using a Multi-Layer Perceptron (MLP) without considering topology to obtain F_{agn}. Simultaneously, we apply the PageRank algorithm to extract the representation of each device based solely on topology, generating F_{ony}. Finally, we concatenate X, F_{agn}, and F_{ony}, and feed them into a graph-attention learner to compute the representations h_t for each temporal instance t. These representations are subsequently processed through a cross-temporal attention module to produce the final prediction.

3.2 Temporal Graph Building

Following prior works, we represent devices as nodes and traffic as edges. In the adjacency matrix A, $A[i, j] = 1$ if there is traffic between devices i and j. The payload of traffic between devices i and j is encoded as a vector, denoted as x_{ij}. By stacking these edge features along the row axis, we obtain the feature matrix X. Notably, in this context, the graph is often directed.

To capture temporal variations in network traffic, we partition the raw data into fixed time windows, constructing multiple graph structures. In each window, devices appearing within the time frame are treated as nodes, and traffic within the window defines the edges. Mathematically, for time window t, the corresponding graph is represented as $\mathcal{G}_t = (A_t, X_t)$, where $A_t[i, j] = 1$ if traffic exists between devices i and j during t, and X_t is the stacked payload information for that window.

3.3 Topology Aggregations

The graph construction process yields a series of graphs, $[\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_t]$. In this subsection, we process each graph individually before integrating temporal information in the next stage.

3.3.1 Topology-agnostic Feature Learner

Previous approaches that disregard graph structures have demonstrated promising results, indicating that payload information alone can be indicative of malicious traffic. To leverage this insight, we first apply an MLP to learn a representation of each payload, offering an alternative perspective for our model. Mathematically, for graph $\mathcal{G}_t$, this process is formulated as:

$$F_{agn}^t = W_1 \cdot Z_t + B_1 \tag{2}$$

where W_1 and B_1 are learnable parameters independent of t.

3.3.2 Topology-only PageRank

In real-world cyberspace, malicious traffic does not always travel directly from the attacker's device to the target. Instead, it may pass through multiple intermediary nodes to evade detection. This increases the need for an extended receptive field. However, the receptive field in Graph Neural Networks (GNNs) is typically constrained by model depth, and increasing depth can lead to over-smoothing. To address this, we propose an alternative approach that encodes topology information without relying on deep GNNs.

Inspired by prior work [16], we utilize PageRank to capture topological characteristics. The computation of PageRank is expressed as:

$$F_{ony}^{t} = (1-d) * \frac{1}{n} * \mathbf{1}_n + d * A_t^T * F_{ony}^{t} \tag{3}$$

where $F_{ony}^{t} \in \mathbb{R}^{|\mathcal{V}|\times 1}$ represents the PageRank values for each node at time t. Here, $\mathbf{1}_n$ is a vector of ones, and d is a fixed hyperparameter set to 0.8. We initialize F_{ony}^{t} by assigning an equal PageRank value of $\frac{1}{|\mathcal{V}|}$ to all nodes and iteratively update F_{ony}^{t} until convergence. For each node v_i, the corresponding PageRank value is denoted as $F_{ony}^{t}[v_i]$. PageRank helps our model understand which nodes are central or influential in **entire** network traffic. This structural knowledge is then combined with **localized** feature learning to improve anomalous traffic detection.

3.3.3 Graph Attention Learner

In the graph attention learner, we compute the attention value for each edge $i \sim j$ based on X_t and F_{ony}^{t} as follows:

$$\varphi_{ij} = \frac{\exp(\partial(W_2 \cdot X_t'[e_{ij}]))}{\sum_{\mathcal{N}_i}^{g}\sum_{\mathcal{N}_j}^{k}[\, W_2 \cdot X_t'[e_{ig}] \,||\, W_2 \cdot X_t'[e_{jk}]\,]} \tag{4}$$

where $X_t'[e_{ij}] = F_{ony}^{t}[v_i]||X_t[e_{ij}]||F_{ony}^{t}[v_j]$, W_2 is a learnable parameter independent of t, and ∂ is an activation function.

The edge representation h_{ij} is then obtained as:

$$h_{ij} = \frac{1}{|\mathcal{N}_i|}\sum_{\mathcal{N}_i}^{g}\varphi_{ig}\cdot W_4 \cdot X_t'[e_{ig}] + \frac{1}{|\mathcal{N}_j|}\sum_{\mathcal{N}_j}^{k}\varphi_{gk}\cdot W_5 \cdot X_t'[e_{gk}] + W_6 \cdot F_{agn}^{t}[e_{ij}] \tag{5}$$

where $\mathcal{N}_i$ denotes the neighbors of v_i, and $W_{4,5,6}$ are learnable parameters.

3.4 Cross-temporal Aggregation

After computing edge embeddings for each temporal graph instance, we concatenate them and apply an attention mechanism to obtain the final representation. Let $\mathbf{h}_t$ denote the stacked h_{ij} values for each edge in $\mathcal{G}_t$. Since each $\mathbf{h}_t$ may have a different length, we pad all embeddings to a common length L, resulting in a matrix $H \in \mathbb{R}^{t \times L}$.

The final embeddings are computed as:

$$Z \in \mathbb{R}^{t \times L} = \text{softmax}\left(\frac{HW^Q(HW^K)^T}{\sqrt{d_k}}\right) \tag{6}$$

where $W^{Q,K,V}$ are learnable parameters.

Applying an $\arg\max(\cdot)$ operator to Z, we compute a cross-entropy loss and perform back-propagation.

4 Experiments

4.1 Setup

Datasets Within our empirical research framework, we have carefully selected two publicly available real-world datasets pertinent to intrusion detection: CIC2017 and DarkNet. These datasets present multi-classification challenges, where the proportion of benign instances significantly outweighs that of malicious attack instances. The datasets inherently exhibit multi-classification complexities, with benign instances occurring at a much higher frequency than attack instances. The classification distribution for each dataset across all categories is outlined in Table 4. Intrusion detection datasets possess distinctive characteristics, and our focus lies in multi-classification rather than binary classification. Each dataset is dominated by a 'normal' class alongside various attack subclasses, whose representation varies from 65.07% to 96.83%, thereby establishing a challenging experimental setting.

Table 1 Distribution of the classes of the selected datasets. All data has different types of abnormalities, except for the data that is labeled as *Normal*

Dataset	Classes (names and %)								
DarkNet	Normal	Audio str	Brows.	Chat	File tr.	Email	P2P	Video str	VOIP
	82.82	9.39	0.19	3.21	1.84	0.41	0.16	0.95	1.03
CIC1017	Normal	BruteF.	DoS	DDoS	Web	Bot	Infiltr.		
	70.61	7.28	1.01	13.04	0.37	5.42	2.27		

Table 2 Results on our selected datasets. Numbers in bold are the higher scores achieved in each pair of models

Dataset	Model	Micro F1	Macro F1
CIC2017	SVM	0.923828	0.604707
	LSTM	0.912148	0.451382
	GAT	0.947266	0.548715
	E-ResGAT	0.947571	0.633182
	Ours	**0.949453**	**0.661166**
DarkNet	SVM	0.738037	0.55763
	LSTM	0.765625	0.643901
	GAT	0.753662	0.633261
	E-ResGAT	0.77832	0.668956
	Ours	**0.785889**	**0.686263**

To prevent data leakage, sender/receiver IPs are excluded, along with features that exhibit a high correlation with labels, as determined using pandas.DataFrame.corr().

Baselines We employ three baseline models: SVM, LSTM [11], GAT [23], and E-ResGAT [4]. E-ResGAT integrates residual learning into the GNN framework by leveraging available graph information. The addition of residual connections serves as a strategy to mitigate the impact of class imbalance, preserving original information and enhancing the performance of minority classes in intrusion detection.

Metrics We evaluate our proposed models and comparative baselines using accuracy (ACC) as a primary metric. In addition to ACC, we assess model performance with the F1-score. However, we refrain from overemphasizing the weighted F1-score at the cost of accuracy in minority classes. To address this, we also consider the macro F1-score, which calculates the average F1-score across all classes, irrespective of their relative proportions, ensuring that minority classes are appropriately represented. Both metrics are computed on the test sets of the respective datasets.

4.2 Comparative Study

The results of the comparison experiments are shown in Table 2. From Table 2, we can draw the following observations: **(i)** *Temporal information alone is not a silver bullet.* Our experimental results indicate that SVM consistently outperforms LSTM. Despite being a temporal model, LSTM exhibits the poorest performance, suggesting that simply leveraging temporal information does not guarantee improved results. This underscores the

Table 3 Results on our selected datasets. Numbers in bold are the higher scores achieved in each pair of models

Dataset	Model	Micro F1	Macro F1
CIC2017	Minus Temporal	0.920898	0.648746
	Minus PageRank	0.917969	0.61709
	FULL	**0.939453**	**0.661166**
DarkNet	Minus Temporal	0.753775	0.586306
	Minus PageRank	0.772461	0.663022
	FULL	**0.785889**	**0.686263**

effectiveness of the SVM approach and highlights the challenges of utilizing temporal dependencies effectively. **(ii)** *Effective data modeling is crucial.* Although our dataset originates from log files, we have structured these logs as graphs in GAT, E-ResGAT, and our proposed model. The superior performance of all graph-based methods demonstrates the importance of data modeling. Different modeling approaches introduce distinct prior knowledge, such as first- and second-order proximity in graphs, which significantly impact the model's effectiveness. **(iii)** *Our model achieves the best performance.* Across both datasets, our model consistently delivers the highest performance, confirming its efficiency and effectiveness in handling intrusion detection tasks.

4.3 Ablation Study

As we mentioned before, the proposed model, which incorporates temporal information and global topology information based on PageRank, provides a basis for our model's efficiency. In this subsection, we experiment on a variant of the model that removes these two modules, and the results are shown in Table 3. From Table 3, we can observe the significance of both temporal information and PageRank. However, their importance varies across different datasets. For instance, in CIC2017, PageRank appears to play a more crucial role than temporal information, whereas in DarkNet, temporal information proves to be more influential. Overall, these results validate the effectiveness of our proposed module.

4.4 Case Study

To better analyze our model's performance during training, we recorded its evaluation metrics at each epoch, as shown in Figure 2. These metrics

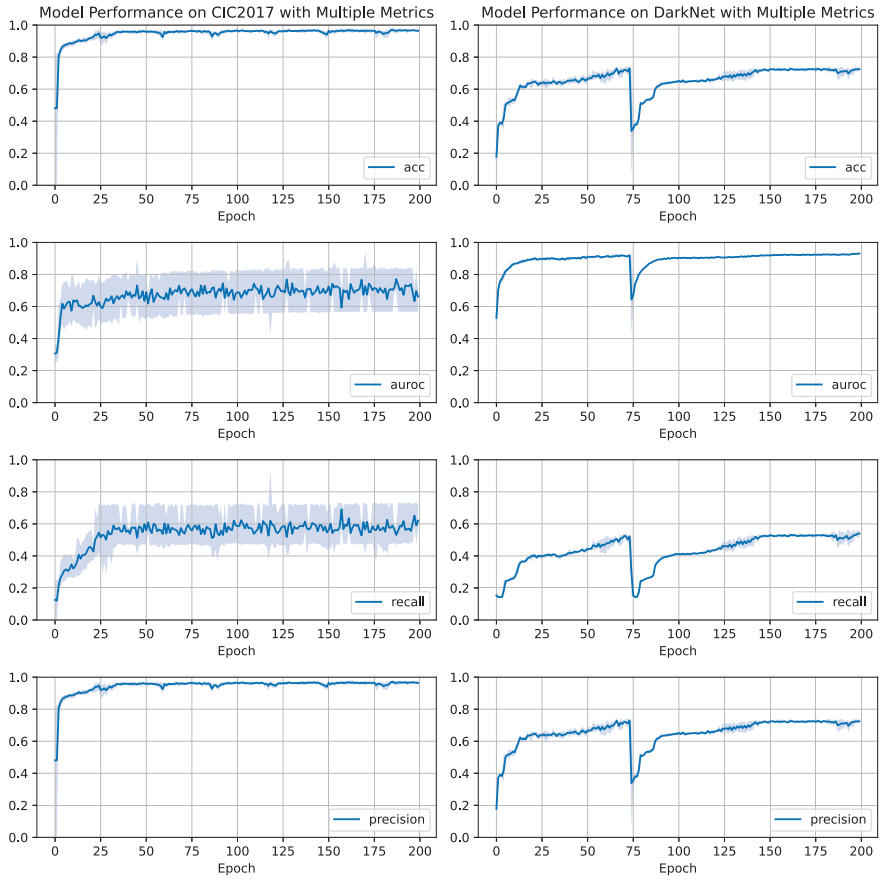

Figure 2 Performance of our model on CIC2027/DarkNet with multiple metrics during the training process, including Accuracy, Area Under the Receiver Operating Characteristic Curve (AORUC), Recall, and Precision.

include Accuracy, Area Under the Receiver Operating Characteristic Curve (AORUC), Recall, and Precision. We also conducted multiple tests under different random seeds, with the corresponding errors visualized in light blue.

We observed that random seeds had a more significant impact on CIC2017, likely due to its larger dataset size and pronounced class imbalance. Variations in random seeds led to different dataset splits and model initialization parameters, resulting in greater performance fluctuations.

Table 4 Quantitative Assessment of Anomaly Type Criticality: Effects of Mislabeling on Darknet Dataset Model Training Performance via Perturbation-Based Interpretability

Clean	Audio str	Brows.	Chat	File tr.	Email	P2P	Video str	VOIP
0.78588	0.61719	0.65723	0.62988	0.63184	0.66602	0.66601	0.64844	0.62891

As shown in Table 1, the DarkNet dataset contains multiple types of anomalies. To investigate which types are more critical, we conducted a perturbation-based model interpretability analysis, a widely used approach in explainable AI [21]. Specifically, we manually relabeled 10% of each anomaly type as *Normal* and then retrain & evaluate the model's performance on this modified dataset. The results are presented in the following table.

The analysis in Table 4 reveals significant differences in how various anomaly types impact model performance. By artificially mislabeling 10% of each anomaly type as "Normal" and examining the resulting performance degradation, we found that "Audio Stream" (Audio str) anomalies had the most significant impact, causing accuracy to drop from 0.78588 to 0.61719. In contrast, "Email" and "Peer-to-Peer" (P2P) anomalies had the least impact, with accuracy decreasing to 0.66602 and 0.66601, respectively. This suggests that some anomaly types are more influential than others, emphasizing the need for targeted handling strategies to improve model robustness. Moreover, the mislabeling of "Audio str" and "Browsing" anomalies had the greatest effect, reducing accuracy to 0.61719 and 0.65723, respectively. This indicates that these anomaly types play a crucial role in distinguishing normal and abnormal behavior. On the other hand, mislabeling the "VOIP" anomaly had the least impact, with accuracy decreasing only slightly to 0.62891. While all anomaly types contribute to model performance, some have a more pronounced effect, highlighting the importance of addressing these variations appropriately.

5 Conclusions

In this paper, we propose a temporal graph neural network designed for anomalous traffic detection. Our model leverages PageRank to encode the positional significance of each node within the overall topology, offering a global structural perspective. Additionally, the temporal information aggregation module, built on an attention mechanism, enhances the model's ability to capture temporal patterns effectively. Extensive experiments on real-world datasets validate the effectiveness of our approach.

References

[1] Majjed Al-Qatf, Lasheng Yu, Mohammed Al-Habib, and Kamal Al-Sabahi. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access*, 6:52843–52856, 2018.

[2] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

[3] Okan Can and Ozgur Koray Sahingoz. A survey of intrusion detection systems in wireless sensor networks. In *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pages 1–6. IEEE, 2015.

[4] Liyan Chang and Paula Branco. Graph-based solutions with residuals for intrusion detection: the modified e-graphsage and e-resgat algorithms. *arXiv preprint*, 2021.

[5] Leyan Deng, Defu Lian, Zhenya Huang, and Enhong Chen. Graph convolutional adversarial networks for spatiotemporal anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2416–2428, 2022.

[6] Ozgur Depren, Murat Topallar, Emin Anarim, and M Kemal Ciliz. An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, 29(4):713–722, 2005.

[7] Bo Dong and Xue Wang. Comparison deep learning method to traditional methods using for network intrusion detection. In *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pages 581–585. IEEE, 2016.

[8] Fahimeh Farahnakian and Jukka Heikkonen. A deep auto-encoder based approach for intrusion detection system. In *ICACT*, 2018.

[9] Guoli Feng, Ning Wang, Xinnan Ha, Xiaobo Li, Run Ma, and Peng Lin. Traffic anomaly detection for smart grid based on hierarchical spatial-temporal feature learning. In *BMSB*, 2023.

[10] Jordan Hochenbaum, Owen S Vallis, and Arun Kejariwal. Automatic anomaly detection in the cloud via statistical learning. *arXiv preprint arXiv:1704.07706*, 2017.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[12] Guanbo Jia, Paul Miller, Xin Hong, Harsha Kalutarage, and Tao Ban. Anomaly detection in network traffic using dynamic graph mining with a sparse autoencoder. In *TrustCom/BigDataSE*, 2019.

[13] Nicolas Keriven. Not too little, not too much: a theoretical analysis of graph (over) smoothing. *Advances in Neural Information Processing Systems*, 35:2268–2281, 2022.

[14] Ae Chan Kim, Mohyun Park, and Dong Hoon Lee. AI-IDS: application of deep learning to real-time web intrusion detection. *IEEE Access*, 8:70245–70261, 2020.

[15] Jihyun Kim, Jaehyun Kim, Huong Le Thi Thu, and Howon Kim. Long short term memory recurrent neural network classifier for intrusion detection. In *International Conference on Platform Technology and Service*, pages 1–5, 2016.

[16] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019.

[17] Zhipeng Li, Zheng Qin, Kai Huang, Xiao Yang, and Shuxiong Ye. Intrusion detection using convolutional neural networks for representation learning. In *International Conference on Neural Information Processing*, pages 858–866. Springer, 2017.

[18] Wei Lin, Chen Li, Li Xu, and Kun Xie. Enhanced network traffic anomaly detection: Integration of tensor eigenvector centrality with low-rank recovery models. *IEEE Transactions on Services Computing*, pages 1–16, 2024.

[19] Dan Niu, Jin Zhang, Li Wang, Kaihong Yan, Tao Fu, and Xisong Chen. A network traffic anomaly detection method based on cnn and xgboost. In *2020 Chinese Automation Congress (CAC)*, pages 5453–5457, 2020.

[20] Shuyu Pei, Jigang Wen, Kun Xie, Gaogang Xie, and Kenli Li. Online network traffic anomaly detection based on tensor sketch. *IEEE Transactions on Parallel and Distributed Systems*, 34(12):3028–3045, 2023.

[21] Luyu Qiu, Yi Yang, Caleb Chen Cao, Yueyuan Zheng, Hilary Ngai, Janet Hsiao, and Lei Chen. Generating perturbation-based explanations with robustness to out-of-distribution data. In *Proceedings of the ACM Web Conference 2022*, pages 3594–3605, 2022.

[22] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint*, 2017.

[24] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Applying convolutional neural network for network intrusion detection. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1222–1228. IEEE, 2017.

[25] Bo Wang, Yang Su, Mingshu Zhang, and Junke Nie. A deep hierarchical network for packet-level malicious traffic detection. *IEEE Access*, 8:201728–201740, 2020.

[26] Wei Wang, Yiqiang Sheng, Jinlin Wang, Xuewen Zeng, Xiaozhou Ye, Yongzhong Huang, and Ming Zhu. HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access*, 6:1792–1806, 2018.

[27] Wenbin Yao, Bangli Pan, Yingying Hou, Xiaoyong Li, and Yamei Xia. An adaptive model filtering algorithm based on grubbs test in federated learning. *Entropy*, 25(5):715, 2023.

[28] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, 5:21954–21961, 2017.

[29] Xingtao Zuo, Cheng Fang, and Ping Han. Network traffic anomaly detection based on spatio-temporal dynamic graph. In *ICEIEC*, 2024.

Biographies

Minghui Gao obtained a Bachelor's degree in Information Security from Northeastern University in 2009. His research areas include information security and network security of power monitoring systems.

Zhijun Zhang graduated from Northeastern University in 1999. His research areas include physical security, specialized security equipment design, artificial intelligence learning, and business data analysis.

Liangliang Cui obtained a Bachelor's degree in Software Engineering from Liaoning Technical University and a Master's degree in Circuits and Systems from Northeastern University. He is responsible for the research and application of network security products for power monitoring systems based on the studies of the Southern Power Grid.

Sibo Feng obtained a Bachelor's degree in Automation from Shenyang University of Technology. His research areas include the security protection

work for grid company master stations, substations, grid-connected power plants, and the safety of power generation side centralized control centers.

Jingyi Liu obtained a Bachelor's degree in Internet of Things Engineering from Shenyang University of Technology. She is currently pursuing a Master's degree in Software Engineering at Northeastern University. Her research interests include artificial intelligence security and 3D computer vision.

Yongzhen Jiang obtained a Bachelor's degree in Software Engineering from Shenyang University of Technology in 2023. She is currently pursuing a Master's degree in Software Engineering at Northeastern University. Her research interests include artificial intelligence security, graph neural networks, and 3D computer vision.