MalDetectFormer: Leveraging Sparse SpatioTemporal Information for Effective Malicious Traffic Detection

Shuai Zhang[1,*,†], **Yu Fan**[2,*], **Haoyi Zhou**[1,2], **Bo Li**[1,2,†]

[1]Zhongguancun Laboratory, Beijing, P.R.China
[2]SKLCCSE, School of Computer Science and Engineering, Beihang University, Beijing, P.R.China
zhangshuai@zgclab.edu.cn, {fanyu, zhouhy, libo}@act.buaa.edu.cn

Abstract

Malicious traffic detection is one of the main challenges in the field of cybersecurity. Although modern deep learning methods have made progress in identifying malicious traffic, they often overlook the persistent nature of attack behaviors, making it difficult to distinguish between malicious and normal traffic at a single observation point. To address this issue, we propose MalDetectFormer, which aims to accurately capture the spatio-temporal dynamics of malicious traffic. By incorporating a sparse attention mechanism, MalDetectFormer can efficiently focus on key characteristics of traffic nodes while overcoming the challenges faced by traditional long-sequence processing. Additionally, by adopting a time-cyclic attention mechanism, the model can identify and capture persistent attack patterns of malicious traffic. Experiments conducted on benchmark datasets demonstrate the advantages of the proposed MalDetectFormer in both malicious traffic detection and malicious attack recognition tasks.

Introduction

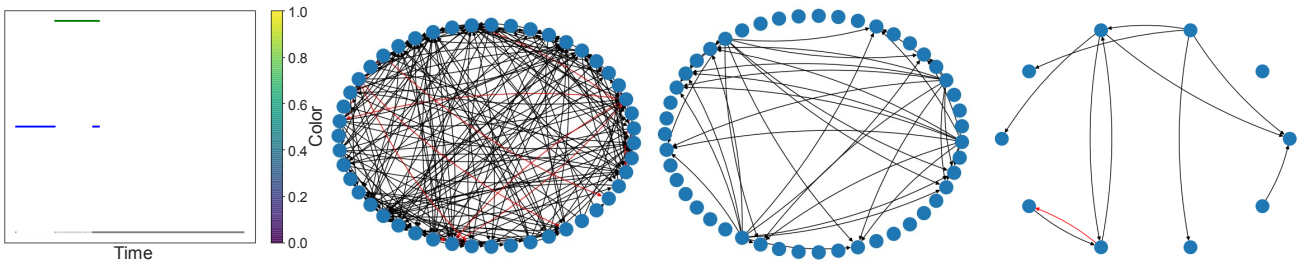
In today's digital era, malicious traffic (Cukier and Panjwani 2007) poses a severe threat to the information security of network systems and users. In the field of cybersecurity, many current methods for detecting malicious traffic rely on predefined rules or signatures to identify known attack patterns or use statistical principles for detection. However, these methods face challenges when encountering unknown attack patterns and have limitations in dealing with the evolving cyber threat landscape. With the rapid development of machine learning technologies (Suga, Okada, and Esaki 2019), applying machine learning methods to the identification of malicious network traffic has become an important trend in cybersecurity research.

The main research direction in malicious traffic detection involves the use of convolutional neural networks (CNN) (Fernando, Xiao, and Spring 2023) (Galinkin 2020) (Chapaneri and Shah 2019) (Gao et al. 2020) (Vega et al. 2020), autoencoders (AE) (Yang et al. 2021), and clustering (Fu et al. 2023) (Diallo and Patras 2021) (Wei et al. 2023) to analyze and identify network traffic characteristics. These methods take into account the combinations of multidimensional features of traffic data to identify malicious activities. However, these feature-based recognition mechanisms primarily focus on the immediate manifestations of traffic characteristics (Galinkin 2020), thereby somewhat neglecting the enduring nature of malicious traffic attacks. Consequently, distinguishing between malicious and normal traffic at a single observation point can be a challenging task. While current methods have made significant strides in recognizing malicious traffic, they still face substantial challenges in capturing and identifying malicious activities with long-term behavioral characteristics.

Significant progress has been made in analyzing time characteristics of malicious traffic attacks through the use of long short-term memory networks (LSTM) (Wang et al. 2020) (Lin, Xu, and Xiao 2022) (Anitha et al. 2023) (Hou et al. 2022) (Li et al. 2023b) and Transformer (Luo et al. 2022) (Shi et al. 2023) models. These methods take a macro perspective by examining sequences of malicious traffic data and incorporating the influence of time factors into feature-based analysis of malicious traffic recognition. However, these approaches have not adequately addressed the challenges associated with long-term sequential issues (Wang, Zeng, and Li 2023) (Wang et al. 2023) (Cao et al. 2023) (Li et al. 2023a), the sparsity of malicious traffic attacks (Yang et al. 2023), and the imbalanced nature of attack data (Lin, Xu, and Xiao 2022). Furthermore, existing strategies often treat each traffic instance as an independent event, thereby overlooking the persistent and similar characteristics of malicious traffic attacks and disregarding the continuity and regularity of malicious traffic behavior patterns.

In terms of spatial analysis, several researchers have focused on exploring the interpretability of traffic data (Liu et al. 2023) (Yuan et al. 2023) (Zhang et al. 2023). For example, Zhang et al. (Zhang et al. 2023) have made notable contributions in this area by integrating an improved Graph Attention Network (GAT) (Veličković et al. 2017) with Long Short-Term Memory Networks (LSTM) (Hochreiter and Schmidhuber 1997). Their approach aims to enhance the model's ability to capture malicious traffic characteristics and understand spatiotemporal features. While their method employs innovative techniques in processing traffic data, such as representing each traffic item as a node in a graph to construct a spatial graph structure based on traffic,

[*]Equal contribution.
[†]Corresponding author.



(a) (b) (c)

Figure 1: (a) Dynamic distribution map of network traffic categories at a specific time in the CICIDS (Sharafaldin, Lashkari, and Ghorbani 2018) dataset. Gray signifies normal traffic, other colors denote attacks. Time is on the horizontal axis. (b) Daily IP traffic visualization in the UNSW-NB15 dataset. Blue dots represent IP nodes, black lines show normal traffic, red lines indicate malicious activity. Left image displays all traffic, right focuses on malicious patterns. (c) Network segment traffic visualization in the UNSW-NB15 dataset. Blue dots symbolize network segment nodes, including all IP nodes. Black lines depict normal traffic, red lines signify malicious traffic between segments.

it does not consider the spatial relationships between attackers and victims. Moreover, it fails to account for the persistent point attack patterns and concentration of malicious traffic attacks in space. So there are two main challenges in modern malicious traffic detection:

Challenge 1: Malicious traffic data often shows sparsity, with lengthy durations and low event density. The imbalanced ratio of malicious activity to normal traffic poses a particular challenge due to the long sequential nature of malicious traffic. Data sparsity and persistence pose major technical hurdles for detecting malicious activities.

Challenge 2: Former-based models leverage efficient attention mechanisms that can focus intensely on input features, capturing complex patterns and details within the data. However, this attention mechanism has inherent limitations in the classification decision process, especially when it comes to potentially giving excessive importance to subtle and non-discriminative variations. In the presence of imbalanced data categories, this tendency not only exacerbates overfitting issues but also weakens the model's generalization ability. Therefore, a key challenge in current research is to adjust and optimize the internal mechanisms of the former-based architecture to mitigate the emphasis on non-discriminative features, improve adaptability, and enhance generalization performance in new scenarios.

To tackle these challenges, we introduce an innovative strategy that integrates the spatiotemporal characteristics of malicious traffic attacks. This strategy involves constructing an external processing environment for the former-based model. The goal is to reduce the resource consumption caused by the model's internal self-attention mechanism and mitigate the issue of excessive focus on non-discriminative information, as depicted in Figure 2. By adopting this approach, we free the former-based model from intensive internal detail analysis and instead enhance its capability to analyze and assess external environmental information. This enables the model to detect malicious traffic from a broader informational perspective, resulting in improved detection accuracy and enhanced generalization ability.

Meanwhile, we conducted an in-depth analysis of malicious network traffic attack patterns, revealing their sparse and persistent characteristics in the temporal dimension. We discovered that malicious traffic attacks extend beyond isolated single-point behaviors and instead exhibit integrated attack patterns with varying durations, as shown in Figure 1(a). In terms of spatial characteristics, malicious traffic attacks demonstrate persistent point attack patterns (Figure 1(b)) and centralization (Figure 1(c)). This means that the sources launching malicious traffic attacks continuously target multiple IP addresses, and the IP nodes executing these attacks tend to cluster within specific network segments.

In this paper, we introduce an innovative model named MalDetectFormer, based on the Transformer architecture. MalDetectFormer is designed to thoroughly analyze the spatiotemporal characteristics of malicious traffic, with the goal of improving the accuracy and efficiency of detection. We have made architectural improvements and conducted extensive experiments, leading to the following contributions:

- Identification of the limited generalization capabilities of the former architecture in classification decisions and the design of MalDetectFormer to enhance generalization performance by incorporating the spatiotemporal characteristics of malicious traffic attacks.
- Discovery of unique spatiotemporal characteristics in malicious traffic attacks and the development of MalDetectFormer to leverage these key features, significantly improving the capability for malicious traffic detection.
- Introduction of a novel attention mechanism in MalDetectFormer to address long sequence issues and persistent attack characteristics in malicious traffic attacks.
- Creation of the Sub-Graph Convolutional Network (SGCN) and its integration into the Encoder part of MalDetectFormer to handle persistent point attack patterns and concentrated attacks in malicious traffic.
- Extensive experiments conducted on large real-world datasets demonstrate MalDetectFormer consistently outperforms SOTA models for malicious traffic detection.

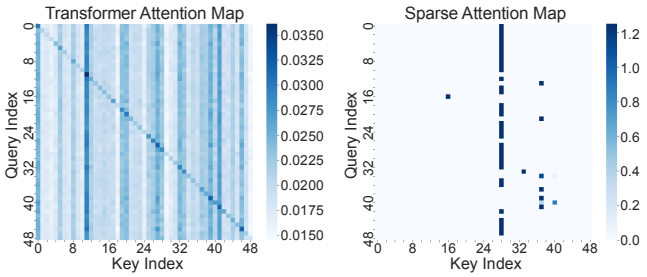


Figure 2: Left: the attention of the multi-head attention mechanism of the Transformer to malicious traffic information. Right: the attention of our sparse multi-head attention to malicious traffic information.

Related Work

Feature-based methods: Fernando et al. (Fernando, Xiao, and Spring 2023) proposed the NeT2I method, which encodes traffic features into images and uses the I2NeT algorithm to decode the images back into traffic for detection and feature extraction. However, this method has limitations in identifying malicious traffic as it already uses rules to sort features during the encoding process. Toki et al. (Suga, Okada, and Esaki 2019) introduced a packet forwarding architecture with an ML classification module, which completes classification during the DNS name resolution process to implement ML-based attack detection. Minghui et al. (Gao et al. 2020) designed a two-tier anomaly detection system based on association analysis and an intrusion detection system. Adrian et al. (Vega et al. 2020) proposed a framework using NetFlow sensors to collect flow datasets and a Docker-based solution designed to easily generate labeled network traffic for building suitable datasets for fitting classification models. Fu et al. (Fu et al. 2023) proposed a novel method called Whisper, which effectively extracts and analyzes the sequential information of network traffic through frequency domain analysis. It performs discrete Fourier transform (DFT) on segmented encoding vectors to extract the traffic sequence information and ultimately uses clustering methods for detection. Yang et al. (Yang et al. 2021) proposed an anomaly detection model with a small number of abnormal samples to address the small-sample detection problem of CNNs and AEs. Diallo et al. (Diallo and Patras 2021) introduced a supervised adaptive clustering (AC) technique to learn cluster centers, which can be used to enhance the feature set of datasets. This method improves the robustness and generalization ability of any classification model against outliers. Nan et al. (Wei et al. 2023) generated clustering features using the k-means algorithm, and the original and generated features were input into a dual classification model constructed with shallow neural networks and random forest algorithms.

Spatial-temporal-based methods: Bo et al. (Wang et al. 2020) proposed a deep layered network for packet-level malicious traffic detection that can effectively identify malicious traffic. Lin et al. (Lin, Xu, and Xiao 2022) utilized the automatic feature extraction capabilities of deep learning to integrate the byte, timing, and short-term statistical features of traffic, employing an Adaptive Balance Training method (ABT) to address the challenges posed by imbalanced data in model training. Anitha et al. (Anitha et al. 2023) combined CNNs with LSTMs, with the proposed technique aimed at reducing the prediction time for identifying malicious traffic by focusing on level prediction to address challenges related to time estimation. Hou et al. (Hou et al. 2022) proposed a malicious traffic detection model based on feature enhancement, grouping the original traffic features according to Gaussian eigenvalues and detecting network traffic using a dual classification model built with shallow neural networks and random forest algorithms. Luo et al. (Luo et al. 2022) introduced a Transformer-based encoder designed to automatically select key features of IoT traffic for detection tasks. Li et al. (Li et al. 2023b) developed a two-tier model using CNN and Bi-SRU for hierarchical feature learning, employing a layered learning approach to learn features of raw traffic data of varying lengths. Shi et al. (Shi et al. 2023) proposed a BERT-based model; the first leverages the attention mechanism to capture global traffic features, while the second constructs a temporal feature extraction module to capture the time-series features of traffic. Beibei et al. (Hu et al. 2019) found a strong correlation between traffic volume and speed, noting distinct morning and evening peak characteristics within a day and significant differences between weekdays and weekends. Qingjun et al. (Yuan et al. 2023) designed a boundary-enhanced method for malicious traffic identification that can more accurately construct decision boundaries and improve accuracy. Zhang et al. (Zhang et al. 2023) combined GATs with LSTM networks, effectively capturing the spatial topological and temporal characteristics of network traffic.

MalDetectFormer

Given the input traffic data $X = \{x_1, x_2, ..., x_T\}$, where $x_t \in R^N$ and N represents the feature dimension of traffic information, the objective is to predict the traffic classification for $Y = \{y_1, y_2, ..., y_T\}$. To tackle the challenges associated with the sparsity and persistence of malicious traffic attacks over time, we re-design the attention mechanism called SparseTimeCycleAttention. This attention mechanism specifically targets spatial patterns of sustained point attacks and concentration. Additionally, we have developed the Sub-Graph Convolutional Network SGCN.

To address these challenges, we have designed the architecture of the MalDetectFormer model, shown in Figure 3. In the following sections, we will introduce the main modules of the model.

Time-Period Embedding

In transformer-based models, the attention mechanism is typically insensitive to the order of inputs, leading to a potential impact on the embedding performance of temporal data. To improve the recognition of temporal changes, previous studies have introduced local positional information and temporal data into the encoding process (Zhou et al. 2021). However, these studies often relied on fixed positional encoding or specific temporal embeddings. When analyzing

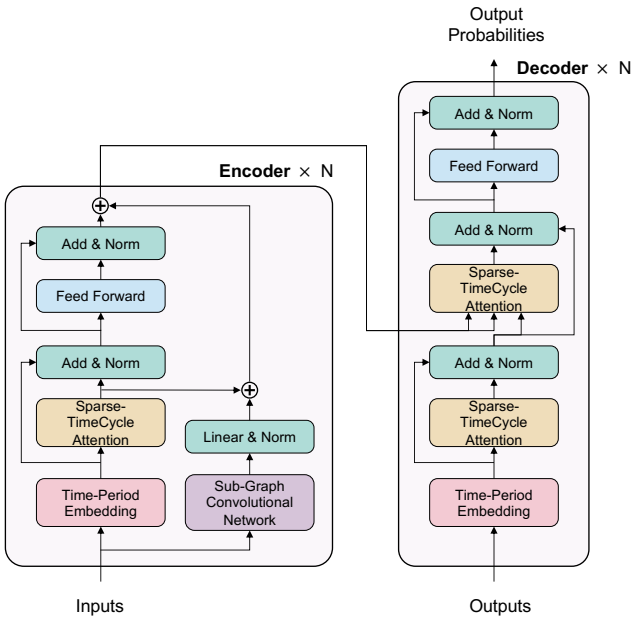


Figure 3: The MalDetectFormer architecture. The left part is the encoder, which applies time-related embeddings before it, and an integration window is used to capture periodic correlations. It also introduces a reinforced graph neural network to capture the spatial aspects of traffic data. The right part is the decoder, used for processing traffic detection.

the patterns of malicious network traffic, which often exhibit periodic characteristics, previous research attempted to capture this periodicity with a vague periodic encoding instead of a straightforward and explicit approach. This approach allows the model to more flexibly recognize and adapt to the temporal variations in attack patterns that possess periodic features.

To address this limitation, we have designed a Time-Period (TP) embedding method, where the dimensions of the position encoding match those of the embedding dimension d_{model}. Additionally, we have introduced an "interval" parameter to finely control the fluctuation cycle within the positional encoding. By incorporating this method, we not only enhance the model's ability to comprehensively capture overall attack patterns but also overcome the constraints of traditional periodic positional encoding for data with significant periodic characteristics. The introduction of variability allows the model to better understand dynamic changes within cycles, thus improving its generalization capability. This fine-grained periodic adjustment provides the model with a more flexible and sensitive mechanism to identify and decipher subtle differences in periodic changes within complex traffic data.

$$
\begin{aligned}
TPE_{(pos,2i)} &= \sin\left(\frac{pos}{interval} + \frac{1}{10000^{2i/d_{model}}}\right), \\
TPE_{(pos,2i+1)} &= \cos\left(\frac{pos}{interval} + \frac{1}{10000^{2i/d_{model}}}\right).
\end{aligned} \tag{1}
$$

Sparse-TimeCycle Attention

Sparse Attention In current research on self-attention mechanisms, the fundamental concept revolves around calculating the similarity between elements within a sequence to capture the intricate dependencies among the data. However, when it comes to detecting malicious traffic, a significant challenge arises due to the sparsity of normal traffic and its substantial presence in datasets. This similarity calculation may inadvertently cause the model to excessively focus on non-representative features, thereby hindering its ability to effectively learn long-time series characteristics. This issue becomes particularly pronounced when dealing with ordinary traffic over extended time spans, as its inherent diversity can prompt the model to learn overly intricate and subtle features, consequently diminishing the model's overall generalization ability.

To tackle this challenge, as shown in the upper part of Figure 4, we propose the integration of a sparsity mechanism into the self-attention mechanism, specifically aimed at reducing the attention allocated by the query vector ($\mathbf{Q}$) to the key vector ($\mathbf{K}$). By adopting this approach, our method not only enables the model to maintain focus on crucial information during the processing of long sequence data but also encourages the learning of more generalized feature representations from a wide spectrum of traffic. This is crucial in preventing the model from being dominated solely by the overwhelming amount of ordinary traffic.

The traffic data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ undergoes distinct linear transformations to derive the Query, Key, and Value for each attention head:

$$
\mathbf{Q}_i^S = \mathbf{X}\mathbf{W}_i^{\mathbf{Q}^S}, \quad \mathbf{K}_i^S = \mathbf{X}\mathbf{W}_i^{\mathbf{K}^S}, \quad \mathbf{V}_i^S = \mathbf{X}\mathbf{W}_i^{\mathbf{V}^S}, \tag{2}
$$

where $i = 1, \ldots, h$, and h represents the total number of heads in multi-head attention. The matrices $\mathbf{W}_i^{\mathbf{Q}^S}, \mathbf{W}_i^{\mathbf{K}^S}$ and $\mathbf{W}_i^{\mathbf{V}^S} \in \mathbb{R}^{d_{model} \times k}$ are learnable parameter matrices, with d_k being the dimension of the query for each head. Then execute `sparsity` operations on the Key and Value.

$$
\begin{aligned}
\mathbf{A}_i^S &= \text{Softmax}\left(\frac{\mathbf{Q}_i^S \mathbf{K}_i^S}{\sqrt{d_k}}\right), \\
\mathbf{Z}_i^S &= \text{Sparse}(\mathbf{A}_i^S \mathbf{V}_i^S), \\
\mathbf{Z}^S &= \text{Contact}(\mathbf{Z}_1^S, \ldots, \mathbf{Z}_h^S) W^s,
\end{aligned} \tag{3}
$$

where $\mathbf{A}_i^S$ represents the attention scores for each head, $\mathbf{Z}_i^S$ is the attention output for each head, $\mathbf{Z}^S$ is the final Sparse attention output, and $\mathbf{W}^S \in \mathbb{R}^{d_k \times d_{model}}$ is a learnable parameter matrix. `Concat` is used to combine the outputs from all heads. `Sparse` refers to the sparsity processing of $\mathbf{Z}_i^S$. Given the structural uniformity of $\mathbf{Z}_i^S$, we emphasize the implementation of consistent sparsity layers to maintain structural and functional integrity. Considering that the model also involves persistent, attack-focused attention calculations, this process inherently leads to a significant increase in computational complexity, especially when handling large-scale datasets. To effectively balance model performance with computational resource consumption, we have designed the

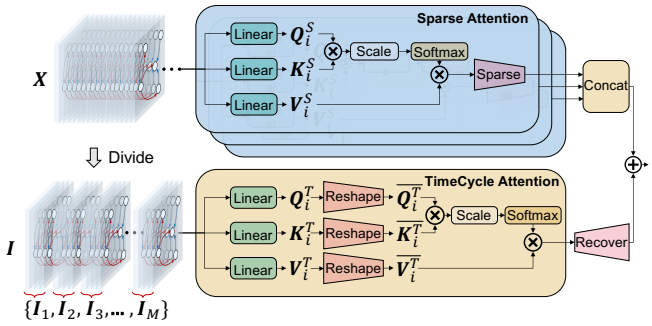


Figure 4: SparseTimeCycleAttention mechanism.

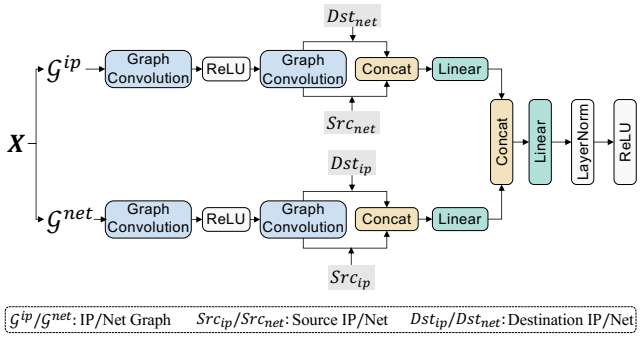


Figure 5: Sub-Graph Convolutional Network.

following sparsity processing mechanism:

$$p(L) = p_0 + \alpha \times \tanh(\beta \times (L - L_{acc})) \tag{4}$$

where L represents the current loss difference, and L_{acc} represents the cumulative average loss difference. α and β are tuneable coefficients, p_0 is the sparsity rate, and $p(L)$ determines the subsequent sparsity rate.

TimeCycle Attention Previously, the detection of malicious traffic relied on analyzing the similarity between individual instances to infer their potential categories. However, this method did not fully account for the persistent attack characteristics of malicious traffic, which indicate that traffic attacks occur continuously over a certain period. Consequently, relying solely on similarity analysis between individual instances might overlook the overall behavioral patterns of malicious traffic, resulting in a limited understanding of its nature.

To address this limitation, as shown in the lower part of Figure 4, we propose a temporal cyclic attention mechanism, which aims to capture the overall process of malicious traffic from a macro perspective. This approach enables more accurate identification and differentiation of periodic malicious attacks. By analyzing malicious traffic over different time intervals, the model gains the ability to recognize attack patterns that recur within specific time windows. As a result, the model focuses not only on the local similarities between traffic instances but also on the temporal patterns throughout the attack process.

The traffic data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ is divided into a series of intervals $\mathbf{I} = \{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_M\}$. Each interval $\mathbf{I}_M$ contains traffic data over a specific period, represented as $\mathbf{I}_m = \{\mathbf{x}_1^m, \mathbf{x}_2^m, ..., \mathbf{x}_{\text{interval}}^m\}$, where interval refers to the size of the time interval, and $\mathbf{x}_i^m$ denotes the i-th traffic data in the m-th interval. To ensure data consistency, a zero-padding strategy is employed to handle tail data that cannot be evenly divided, thereby ensuring uniform data dimensions across all intervals.

$$\mathbf{Q}_i^t = \mathbf{I}\mathbf{W}_i^{\mathbf{Q}^t}, \quad \mathbf{K}_i^t = \mathbf{I}\mathbf{W}_i^{\mathbf{K}^t}, \quad \mathbf{V}_i^t = \mathbf{I}\mathbf{W}_i^{\mathbf{V}^t}, \tag{5}$$

When faced with the raw data format, directly applying the standard multi-head attention mechanism would compute similarities within intervals rather than capturing the inter-interval similarities that we aim to capture. This limitation arises from the core operation of the attention mechanism, which involves the dot product between the Query (**Q**) and Key (**K**), reflecting the correlation between input features. Therefore, this study proposes a specific formatting strategy for processing **Q** and **K** to ensure that the attention scores accurately reflect the similarity between time intervals. The multi-dimensional features of each time interval in the raw data are transformed into a single vector representation that encapsulates the overall characteristics of that interval. Specifically, **Q**, **K**, and **V** are first reshaped according to the interval, integrating and compressing all features of each interval into a dimension-flattened vector. This allows subsequent dot product operations to extend beyond individual time points or features and encompass the global characteristics of the entire interval.

$$\overline{\mathbf{Q}^t} = \text{Reshape}(\mathbf{Q}^t), \overline{\mathbf{K}^t} = \text{Reshape}(\mathbf{K}^t), \overline{\mathbf{V}^t} = \text{Reshape}(\mathbf{V}^t) \tag{6}$$

The `Reshape` operation converts **Q** and **K** from multi-dimensional data into two-dimensional data that includes "interval" dimension indices and their composite feature vectors.

$$\mathbf{Z}_c^t = \text{Softmax}(\frac{\overline{\mathbf{Q}^t} \cdot \overline{\mathbf{K}^t}}{\sqrt{d_k}})\mathbf{V}^t, \quad \mathbf{Z}^t = \text{Recover}(\mathbf{Z}_c^t) \tag{7}$$

where $\mathbf{Z}_c^t$ represents the result of the computation of periodic similarity. The subsequent `Recover` operation restores $\mathbf{Z}_c^t$ to the same dimensional as $\mathbf{Z}^s$, ensuring the structural integrity and comparability of the data across different computational stages. The final output is denoted as $\mathbf{Z}^t$, representing the TimeCycle output.

Sub-Graph Convolutional Network

One of the fundamental challenges in constructing graph models for network traffic analysis is effectively representing and utilizing spatial information within the graph, particularly when traffic information primarily exists as attributes of edges rather than being directly associated with nodes. This necessitates the adoption of refined strategies during the graph construction process to ensure that the dynamic characteristics of network traffic and their spatial distribution within the graph structure are fully captured and expressed.

Problem Formulation The traffic graph structure $\mathcal{G}^{ip} = \{\mathcal{N}^{ip}, \mathcal{E}^{ip}, \mathcal{V}^{ip}, \mathcal{A}^{ip}\}$ consists of a set of network nodes $\mathcal{N}^{ip}$

and edges $\mathcal{E}^{ip}$, corresponding to relationships between different nodes. $\mathcal{V}^{ip}$ denotes the attributes of the nodes, while $\mathcal{E}^{ip}$ represents the attributes of the edges. To effectively capture the spatial attributes of traffic attacks, we construct a set of nodes from the source IP Src_{ip} and the destination IP Dst_{ip} of each traffic information.

$$\mathcal{N}_i^{ip} = \{\text{Src}_i^{ip}, \text{Dst}_i^{ip}\}, \mathcal{N}^{ip} = \cup(\mathcal{N}_1^{ip}||\mathcal{N}_2^{ip}||...||\mathcal{N}_T^{ip}), \quad (8)$$

where $\cup$ represents the integrated node information of $\mathcal{E}_i^{ip}$.

When constructing node attributes, we choose to retain only the spatial attributes of the nodes and exclude the remaining attributes of the traffic information. Including additional traffic information within the nodes could contaminate the spatial information, as characteristics such as packet size do not belong to the spatial attributes of the nodes, $\mathcal{V}_{id}^{ip} = \{\text{IP}_{id}, \text{Net}_{id}\}$. Communication between two nodes occurs multiple times, rather than just once, resulting in recurrent edges (traffic communications) between nodes. This introduces inconsistencies in information, as communication between networks cannot be averaged like traffic flow to obtain the attributes of a single edge. Such an approach would only introduce complexity and volatility, failing to accurately reflect the spatial characteristics of the nodes. Therefore, we set $\mathcal{A}^{ip} = \emptyset$. Considering that malicious traffic attacks often exhibit significant concentration in cyberspace, we introduce the concept of the subnet graph $\mathcal{G}^{net} = \{\mathcal{N}^{net}, \mathcal{E}^{net}, \mathcal{V}^{net}, \mathcal{A}^{net}\}$. In this graph, network subnets are represented as nodes, and the communication relationships between these subnets serve as edges, forming a novel graphical representation. This approach enables the model to analyze the behavior of individual IP addresses at the node level, while at a higher level of abstraction, the subnet level, it captures the spatial concentration phenomena in network communication patterns.

$$\mathcal{N}^{net} = \{\mathcal{N}_i^{net} \big| \forall \text{IP}_m \in \mathcal{N}^{ip}, \exists \int_i \in \mathcal{S} : f(\text{IP}_m) = \int_i\} \quad (9)$$

for each IP address $\text{IP}_m \in \mathcal{N}^{ip}$, there exists a mapping function f that maps the IP_m to the subnet $\int_i, f : \mathcal{N}^{ip} \to \mathcal{S}$, where $\mathcal{S}$ is the set of subnets.

$$\mathcal{E}^{net} = \{(\mathcal{N}_i^{net}, \mathcal{N}_j^{net}) \big| \exists \text{IP}_m \in \mathcal{N}_i^{net}, \exists \text{IP}_n \in \mathcal{N}_j^{net} : \text{IP}_m \leftrightarrow \text{IP}_n\} \quad (10)$$

where $\text{IP}_m \leftrightarrow \text{IP}_n$ indicates that there is communication between IP_m and IP_n.

$$\mathcal{V}_{id}^{net} = \{\overline{\text{IP}_{id}}, \text{Net}_{id}\} \quad (11)$$

where $\overline{\text{IP}_{id}}$ represents the average value of the IP addresses within the subnet and we set $\mathcal{A}^{net} = \emptyset$. We use SGCN to capture the spatiality of the model, as shown in Figure 5.

Experiment

Experiment Setting and Results

We evaluate the performance of MalDetectFormer* with malicious traffic detection tasks in three dimensions: (1) in non-realistic environments, we assess the ability of different models to recognize and analyze the characteristics of malicious traffic; (2) we explore the capability of models to handle homogenous attacks involving multiple attack strategies in environments close to realistic settings; (3) we investigate the detection effectiveness of models in real environments where the attacks contain multiple types of malicious traffic.

Meanwhile, we conducted ablation experiments during the experiment to investigate the impact of time and space modules on model performance. our_t represents using only Sparse-TimeCycle Attention to explore the impact of time modules on model performance. our_s represents using only Sub-Graph Convolutional Network to explore the impact of space modules on model performance.

Malicious Traffic Detection for Single Attack Type In this section, we explored the model's detection capabilities when dealing with the same type of attack that involves multiple attack methods. For this purpose, we utilized the CIC-IDS-2017 dataset, which covers a diverse range of complex attack scenarios. Specifically, we conducted exhaustive experiments on each individual file within this dataset to assess the model's response to different attack strategies.

For files that include multiple attack methods, we configured 40% of the data as the training set, 10% for the validation set to tune the model parameters, and the remaining 50% as the test set to evaluate the model's ability to recognize complex attack patterns. For files containing only a single type of attack, the dataset was divided into 70% training, 10% validation, and 20% test sets, aimed at verifying the model's effectiveness in simpler scenarios. For specific files with sparse attack methods, such as WebAttack and MultiDos, we adopted a single-type attack detection strategy to ensure detection accuracy and avoid model performance degradation due to data sparsity. The experimental results are shown in Table 1 below.

In the malicious traffic attack test close to the real environment, it can be observed that the performance of all baseline models generally declines, especially the model based on the Former architecture. This finding validates our preliminary viewpoint that the Former architecture faces challenges in dealing with malicious traffic detection. Despite this, most models can still achieve a high level of convergence when facing a single type of malicious traffic attack. In addition, our model demonstrates its superior performance after considering the characteristics of malicious traffic attacks.

Malicious Traffic Detection for Multiple Attack Types In this section, we consider the most realistic scenario, where nine types of attacks occur simultaneously (fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms), to conduct accurate malicious traffic detection. For this task, we use the UNSW-NB15 dataset, allocating 70% of the data as the training set, 10% as the validation set, and 20% as the test set. The experimental results are shown in Table 2 below.

In the face of the most complex and real-world malicious traffic attack scenarios, we significantly observed further degradation in the performance of all benchmark models, while our model exhibited superior performance. Consider-

*https://github.com/WYuJS/MalDetectFormer

Dataset		LSTM	CNN	BiLSTMCNN	Transformer	Informer	Autoformer	FEDFormer	GMM	$ours_t$	$ours_s$	ours
PortScan	Accuracy	0.9825	0.6174	0.9836	0.6174	0.6174	0.5991	0.7101	0.0932	0.9170	0.9309	**0.9993**
	Precison	0.9828	0.6174	0.9840	0.3812	0.6174	0.4103	0.8005	0.0509	0.9266	0.9377	**0.9993**
	Recall	0.9825	0.6174	0.9837	0.6174	0.6174	0.5991	0.7101	0.0932	0.9170	0.9309	**0.9993**
	F1	0.9824	0.4714	0.9836	0.4714	0.4714	0.487	0.6498	0.0655	0.9146	0.9294	**0.9993**
	AUC	0.9978	0.4999	0.9969	0.5000	0.4976	0.6494	0.7923	0.1215	0.9876	0.9917	**0.9999**
DDOS	Accuracy	0.5734	0.9993	0.9989	0.4265	0.4265	0.7847	0.8715	0.3660	0.9931	0.9925	**0.9994**
	Precision	0.3288	0.9993	0.9989	0.1820	0.1820	0.7949	0.8991	0.3769	0.9932	0.9926	**0.9994**
	Recall	0.5734	0.9993	0.9989	0.4266	0.4266	0.7848	0.8715	0.3630	0.9931	0.9925	**0.9994**
	F1	0.4180	0.9993	0.9989	0.2551	0.2551	0.7860	0.8719	0.3450	0.9931	0.9925	**0.9994**
	AUC	0.4793	0.9999	0.9998	0.3839	0.4996	0.8781	0.9844	0.3632	0.9998	0.9992	**0.9999**
Bot	Accuracy	0.9926	**0.9993**	0.9972	0.9926	0.9926	0.5815	0.9925	0.2449	0.9913	0.9919	0.9991
	Precision	0.9853	**0.9994**	0.9980	0.9853	0.9853	0.9860	0.9902	0.9637	0.9953	0.9953	0.9992
	Recall	0.9926	**0.9994**	0.9972	0.9926	0.9926	0.5815	0.2449	0.9637	0.9913	0.9919	0.9991
	F1	0.9890	**0.9994**	0.9975	0.9890	0.9890	0.7298	0.2449	0.3906	0.9983	0.9986	0.9991
	AUC	0.9642	0.9999	0.8858	0.4915	0.5560	0.6485	0.2451	0.8654	0.9932	0.9936	**0.9999**
Infiltration	Accuracy	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9998	0.7725	0.9999	0.9999	**0.9999**
	Precision	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	**0.9999**
	Recall	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9998	0.7725	0.9999	0.9999	**0.9999**
	F1	0.9999	0.9999	0.9999	0.9999	0.9999	0.1185	0.9999	0.8716	0.9999	0.9999	**0.9999**
	AUC	0.9962	0.9863	0.9410	0.4836	0.6755	0.7430	0.9479	0.8862	0.9968	0.9970	**0.9979**
WebAttacks	Accuracy	0.9911	0.9874	**0.9943**	0.9911	0.9911	0.9886	0.9922	0.2105	0.9941	0.9941	0.9941
	Precision	0.9823	0.9904	0.9922	0.9823	0.9823	0.9886	0.9909	0.9533	0.9921	0.9921	**0.9923**
	Recall	0.9911	0.9874	0.9943	0.9911	0.9911	**0.9986**	0.9923	0.2105	0.9941	0.9941	0.9941
	F1	0.9867	0.9883	0.9928	0.9867	0.9867	**0.9975**	0.9911	0.3447	0.9926	0.9926	0.9927
	AUC	0.9347	0.9746	**0.9937**	0.6167	0.5857	0.8388	0.8752	0.3015	0.9939	0.9965	0.9681
Patator	Accuracy	0.9770	0.9810	0.9791	0.9810	0.9810	0.9810	0.9810	0.6746	0.9810	0.9810	**0.9811**
	Precision	0.9625	0.9624	0.9625	0.9624	0.9624	0.9624	0.9626	0.9673	0.9624	0.9624	**0.9718**
	Recall	0.9770	0.9810	0.9810	0.9810	0.9810	0.9810	0.9810	0.6746	0.9810	0.9810	**0.9811**
	F1	0.9697	0.9717	0.9707	0.9717	0.9717	0.9717	0.9717	0.7948	0.9717	0.9717	**0.9764**
	AUC	0.9824	0.9946	0.9652	0.4931	0.5544	0.5458	0.7674	0.4779	0.9975	0.9150	**0.9993**
Multi Dos	Accuracy	0.9694	0.9724	0.984	0.9609	0.9609	0.9605	0.9609	0.5788	0.9999	0.9999	**0.9999**
	Precision	0.9702	0.9704	0.9835	0.9234	0.9234	0.9246	0.9234	0.9271	0.9999	0.9999	**0.9999**
	Recall	0.9694	0.9724	0.984	0.9609	0.9609	0.9605	0.9609	0.5788	0.9999	0.9999	**0.9999**
	F1	0.9698	0.9662	0.9837	0.9418	0.9418	0.9416	0.9418	0.7018	0.9999	0.9999	**0.9999**
	AUC	0.9764	0.9870	0.9845	0.5001	0.5186	0.4577	0.4661	0.4859	0.9999	0.9999	**0.9999**

Table 1: Anomaly detection on CIC-IDS-2017 dataset

Dataset		LSTM	CNN	BiLSTMCNN	Transformer	Informer	Autoformer	FEDFormer	GMM	$ours_t$	$ours_s$	ours
UNSW-NB15	Accuracy	0.8341	0.8325	0.7353	0.8330	0.8077	0.8330	0.8329	0.8267	0.9772	0.9813	**0.9857**
	Precision	0.8570	0.7168	0.4197	0.6939	0.5008	0.6939	0.7329	0.8816	0.9776	**0.9815**	0.9671
	Recall	0.8341	0.8325	0.9738	0.8330	0.8601	0.8330	0.8330	0.8267	0.9772	**0.9813**	0.9588
	F1	0.7598	0.7570	0.5865	0.7571	0.6330	0.7551	0.7571	0.8426	0.9766	**0.9810**	0.9629
	AUC	0.8906	0.5048	0.8751	0.6543	0.8980	0.7484	0.5121	0.8281	0.9940	0.9944	**0.9985**

Table 2: Anomaly detection on UNSW-NB15 dataset

ing the time sparsity of malicious traffic and the diverse attack types it contains, which increase the complexity, traditional models based on the former often struggle to achieve optimal generalization in this environment. To address this, we introduced a sparse attention mechanism that fully considers the characteristics of malicious traffic and integrates spatial information with SGCN, thereby significantly improving the model's generalization ability

Conclusion

We propose MalDetectFormer for detecting malicious traffic that leverages sparse spatiotemporal information. This approach considers the temporal sparsity and periodicity of attacks, as well as the persistence of point attack patterns and spatial concentration. It effectively addresses the challenges posed by long sequences in malicious traffic detection. Moreover, we demonstrated that incorporating external information resolves potential flaws in the decision-making process of the Former architecture, reducing resource consumption caused by internal self-focus and alleviating excessive attention to non-discriminative information. MalDetectFormer significantly outperforms other models on real-world tasks. This highlights the effectiveness of our method in leveraging unique sparse spatiotemporal attack dynamics for more efficient cybersecurity detection capabilities.

Acknowledgements

This work was supported by the National Science and Technology Major Project (No. 2022ZD0117800), the grants from the Natural Science Foundation of China (62202029), and Young Elite Scientists Sponsorship Program by CAST (No. 2023QNRC001). Thanks for the computing infrastructure provided by Beijing Advanced Innovation Center for Big Data and Brain Computing. This work was also sponsored by CAAI-Huawei MindSpore Open Fund.

References

Anitha, T.; Aanjankumar, S.; Poonkuntran, S.; and Nayyar, A. 2023. A Novel Methodology for Malicious Traffic Detection in Smart Devices Using BI-LSTM-CNN-dependent Deep Learning Methodology. *Neural Comput. Appl.*, 35(27): 20319–20338.

Cao, H.; Huang, Z.; Yao, T.; Wang, J.; He, H.; and Wang, Y. 2023. InParformer: Evolutionary Decomposition Transformers with Interactive Parallel Attention for Long-Term Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6906–6915.

Chapaneri, R.; and Shah, S. 2019. Detection of Malicious Network Traffic using Convolutional Neural Networks. In *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT*, 1–6.

Chen, M.; Peng, H.; Fu, J.; and Ling, H. 2021. AutoFormer: Searching Transformers for Visual Recognition. In *2021 International Conference on Computer Vision, ICCV*, 12250–12260.

Cukier, M.; and Panjwani, S. 2007. A Comparison between Internal and External Malicious Traffic. In *The 18th IEEE International Symposium on Software Reliability, ISSRE*, 109–114.

Diallo, A. F.; and Patras, P. 2021. Adaptive Clustering-based Malicious Traffic Classification at the Network Edge. In *40th IEEE Conference on Computer Communications, INFOCOM*, 1–10.

Fernando, O. A.; Xiao, H.; and Spring, J. 2023. New Algorithms for the Detection of Malicious Traffic in 5G-MEC. In *IEEE Wireless Communications and Networking Conference, WCNC*, 1–6.

Fu, C.; Li, Q.; Shen, M.; and Xu, K. 2023. Frequency Domain Feature Based Robust Malicious Traffic Detection. *IEEE/ACM Trans. Netw.*, 31(1): 452–467.

Galinkin, E. 2020. Malicious Network Traffic Detection via Deep Learning: An Information Theoretic View. *arXiv e-prints*, arXiv:2009.07753.

Gao, M.; Ma, L.; Liu, H.; Zhang, Z.; Ning, Z.; and Xu, J. 2020. Malicious Network Traffic Detection Based on Deep Neural Networks and Association Analysis. *Sensors*, 20(5): 1452.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hou, J.; Liu, F.; Lu, H.; Tan, Z.; Zhuang, X.; and Tian, Z. 2022. A Novel Flow-vector Generation Approach for Malicious Traffic Detection. *J. Parallel Distributed Comput.*, 169(C): 72–86.

Hu, B.; Lin, D.; Sun, Q.; and Dong, X. 2019. Research on Road Traffic Flow Status Based on Survival Analysis. *Journal of Physics: Conference Series*, 1187(5): 052056.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 1746–1751.

Kumar, V.; Sinha, D.; Das, A. K.; Pandey, S. C.; and Goswami, R. T. 2020. An Integrated Rule Based Intrusion Detection System: Analysis on UNSW-NB15 Data Set and The Real Time Online Dataset. *Cluster Computing*, 23(2): 1397–1418.

Li, Y.; Qi, S.; Li, Z.; Rao, Z.; Pan, L.; and Xu, Z. 2023a. SMARTformer: Semi-Autoregressive Transformer with Efficient Integrated Window Attention for Long Time Series Forecasting. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI*, 2169–2177.

Li, Z.; Cheng, Z.; Zang, T.; and Li, Y. 2023b. MTCD-Model: A Two-Layer Model for Malicious Traffic Classification and Detection Based on Hierarchical Feature Learning. In *International Joint Conference on Neural Networks, IJCNN*, 1–8.

Lin, K.; Xu, X.; and Xiao, F. 2022. MFFusion: A Multilevel Features Fusion Model for Malicious Traffic Detection Based on Deep Learning. *Computer Networks*, 202(C): 108658.

Liu, J.; Wang, L.; Hu, W.; Gao, Y.; Cao, Y.; Lin, B.; and Zhang, R. 2023. Spatial-Temporal Feature with Dual-Attention Mechanism for Encrypted Malicious Traffic Detection. *Security and Communication Networks*, 2023: 1 – 13.

Luo, Y.; Chen, X.; Ge, N.; Feng, W.; and Lu, J. 2022. Transformer-Based Malicious Traffic Detection for Internet of Things. In *IEEE International Conference on Communications, ICC*, 4187–4192.

Moustafa, N.; and Slay, J. 2015. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). In *2015 Military Communications and Information Systems Conference, MilCIS*, 1–6.

Sharafaldin, I.; Lashkari, A. H.; and Ghorbani, A. A. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP*, 108–116.

Sharafaldin, I.; Lashkari, A. H.; Hakak, S.; and Ghorbani, A. A. 2019. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In *2019 International Carnahan Conference on Security Technology, ICCST*, 1–8.

Shi, Z.; Luktarhan, N.; Song, Y.; and Yin, H. 2023. TSFN: A Novel Malicious Traffic Classification Method Using BERT and LSTM. *Entropy*, 25(5): 821.

Suga, T.; Okada, K.; and Esaki, H. 2019. Toward Real-time Packet Classification for Preventing Malicious Traffic by Machine Learning. In *22nd Conference on Innovation in*

Clouds, Internet and Networks and Workshops, ICIN, 106–111.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NeurIPS*, 6000–6010.

Vega, A. C.; Crespo-Martínez, I. S.; Higueras, Á. M. G.; and Llamas, C. F. 2020. Flow-Data Gathering Using NetFlow Sensors for Fitting Malicious-Traffic Detection Models. *Sensors*, 20(24): 7294.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2017. Graph Attention Networks. *arXiv e-prints*, arXiv:1710.10903.

Wang, B.; Su, Y.; Zhang, M.; and Nie, J. 2020. A Deep Hierarchical Network for Packet-Level Malicious Traffic Detection. *IEEE Access*, 8: 201728–201740.

Wang, C.; Li, Y.; Sun, X.; Wu, Q.; Wang, D.; and Huang, Z. 2023. DeLELSTM: Decomposition-based Linear Explainable LSTM to Capture Instantaneous and Long-term Effects in Time Series. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI*, 4299–4307.

Wang, L.; Zeng, L.; and Li, J. 2023. AEC-GAN: Adversarial Error Correction GANs for Auto-Regressive Long Time-Series Generation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, 10140–10148.

Wei, N.; Yin, L.; Zhou, X.; Ruan, C.; Wei, Y.; Luo, X.; Chang, Y.; and Li, Z. 2023. A Feature Enhancement-based Model for The Malicious Traffic Detection with Small-scale Imbalanced Dataset. *Inf. Sci.*, 647(C): 119512.

Yang, F.; Li, X.; Wang, M.; Zang, H.; Pang, W.; and Wang, M. 2023. WaveForM: Graph Enhanced Wavelet Learning for Long Sequence Forecasting of Multivariate Time Series. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, 10754–10761.

Yang, H.; He, Q.; Liu, Z.; and Zhang, Q. 2021. Malicious Encryption Traffic Detection Based on NLP. *Secur. Commun. Networks*, 2021: 9960822.

Yuan, Q.; Liu, C.; Yu, W.; Zhu, Y.; Xiong, G.; Wang, Y.; and Gou, G. 2023. BoAu: Malicious traffic detection with noise labels based on boundary augmentation. *Comput. Secur.*, 131: 103300.

Zhang, L.; Tan, L.; Shi, H.; Sun, H.; and Zhang, W. 2023. Malicious Traffic Classification for IoT based on Graph Attention Network and Long Short-Term Memory Network. In *24st Asia-Pacific Network Operations and Management Symposium, APNOMS*, 54–59.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *International Conference on Machine Learning, ICML*, 27268–27286.