EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

DTT: A Dual-domain Transformer model for Network Intrusion Detection

Chenjian Xu[1], Weirui Sun[2] and Mengxue Li[3,*]

[1]Informatization Center, Zhengzhou Information Engineering Vocational College, Zhengzhou 451000, Henan, China
[2]Informatization Center, Ludong University, Yantai 264100, Shandong, China
[3]Shool of Information, Zhengzhou Information Engineering Vocational College, Zhengzhou 451000, Henan, China

Abstract

With the rapid evolution of network technologies, network attacks have become increasingly intricate and threatening. The escalating frequency of network intrusions has exerted a profound influence on both industrial settings and everyday activities. This underscores the urgent necessity for robust methods to detect malicious network traffic. While intrusion detection techniques employing Temporal Convolutional Networks (TCN) and Transformer architectures have exhibited commendable classification efficacy, most are confined to the temporal domain. These methods frequently fall short of encompassing the entirety of the frequency spectrum inherent in network data, thereby resulting in information loss. To mitigate this constraint, we present DTT, a novel dual-domain intrusion detection model that amalgamates TCN and Transformer architectures. DTT adeptly captures both high-frequency and low-frequency information, thereby facilitating the simultaneous extraction of local and global features. Specifically, we introduce a dual-domain feature extraction (DFE) block within the model. This block effectively extracts global frequency information and local temporal features through distinct branches, ensuring a comprehensive representation of the data. Moreover, we introduce an input encoding mechanism to transform the input into a format suitable for model training. Experiments conducted on two distinct datasets address concerns regarding data duplication and diverse attack types, respectively. Comparative experiments with recent intrusion detection models unequivocally demonstrate the superior performance of the proposed DTT model.

Keywords: Network Intrusion Detection, Dual-domain Feature Extraction, Temporal Convolutional Networks, Input Encoding

Received on 17 March 2024, accepted on 24 April 2024, published on 06 May 2024

doi: 10.4108/eetsis.5445

1. Introduction

In contemporary information society, the ubiquitous nature of computer networks and the Internet has enhanced the daily lives of individuals. However, this progress has also led to an increase in the frequency and complexity of network intrusions, posing cybersecurity threats [1]. Major incidents like the Capital One data breach [2] and the SolarWinds attack [3] underscore the urgency of addressing cybersecurity issues. To tackle this challenge, network security professionals utilize advanced technologies to detect malicious network traffic [4]. They aim to swiftly identify and thwart potential attacks to safeguard cyberspace integrity. Consequently, developing efficient and accurate models for network intrusion detection (NID) has emerged as a pivotal research focus within the field of network security.

Early intrusion detection research primarily relied on recognized attack patterns or characteristics, such as rule- and signature-based approaches [5]. However, attackers continuously refine their strategies to evade detection. This poses challenges for traditional methods, which are unable to proactively detect new and emerging threats, such as zero-day vulnerabilities [6]. To address this challenge,

[*] Corresponding author. Email: ly0446@hati.edu.cn

cybersecurity researchers are enhancing intrusion detection systems (IDS) by incorporating machine learning, deep learning, and behavioral analysis [7].

Models like Transformer and Temporal Convolutional Networks (TCN) demonstrate exceptional sequence modeling capabilities, making them well-suited for processing network traffic data [8, 9]. Researchers have widely adopted these models as tools for intrusion detection, leading to significant research advancements. For instance, Liang et al. [10] introduced a multi-level intrusion detection model based on Transformer, which demonstrates remarkable performance in detecting intrusion actions. Cheng et al. [11] proposed a global attentional TCN-based IDS for in-vehicle applications. However, they only utilized one of these methods. We believe that combining the attention mechanism inherent in Transformer with the long-term dependency-capturing capability of TCN could further enhance the performance of IDS.

Current intrusion detection methods based on TCN and Transformer typically explore the time domain while overlooking the frequency domain. Fourier analysis reveals that these models exhibit learning biases toward specific types of frequency components. They fail to perceive the entire spectrum of the network time series data, leading to information loss [12, 13]. However, specific intrusions may involve anomalous traffic patterns or frequent periodic signals within certain frequency ranges. For instance, Fu et al. [14] demonstrated the efficacy of leveraging frequency domain analysis for robust detection. Therefore, the inability to comprehensively extract information from the frequency domain poses a challenge in accurately identifying specific attack types.

Additionally, in terms of input coding, certain studies focus exclusively on categorized fields. They neglect numeric fields [15] or resort to simplistic encoding techniques like one-hot coding for the categorized fields [16]. Consequently, crucial feature information is lost during the data processing stage, potentially compromising the effectiveness of IDS. Inspired by [17], we proposed a flow-level projection (FLP) encoding method to combine categorized and numeric fields.

In light of these limitations, this paper introduces DTT, a Dual-domain intrusion detection model based on TCN and Transformer. The DTT model bridges the gap between time and frequency domain analysis and adopts robust input coding strategies. This addresses the shortcomings of existing intrusion detection methods, particularly in capturing comprehensive feature information and enhancing model performance. The model comprises three modules: FLP encoding mudule, Dual-domain Feature Extraction (DFE) Block, and TCN. The FLP encodes the categorical and numerical fields together, aiming to preserve feature information in the data more effectively. The DFE block extracts both time and spectral domain information to learn high-frequency and low-frequency details. It enables comprehensive analysis of network traffic data, thereby enhancing the model's detection capabilities. The TCN module focuses on capturing long-term dependencies, thus supplementing global features. Experimental results demonstrate the superiority of the DTT model over recent intrusion detection methods on two public datasets. The contributions of this paper are summarized below.

- We introduce a novel intrusion detection model termed DTT. We construct the DFE block based on Transformer. This block extracts multi-level frequency information to capture both global and local features. Additionally, the model integrates TCN to capture long-term dependencies in the network data, thereby improving computational efficiency.

- We integrate the FLP encoding module into the model. It improves the model's ability to comprehensively capture data feature information, subsequently enhancing its overall performance.

- We perform comprehensive experiments to assess the performance of DTT using two extensive datasets. Experimental results showcase the superior performance of the DTT model compared to recent intrusion detection models.

The paper comprises the following sections: section 1 provides an introduction to our research, section 2 explores recently published related works, section 3 outlines the proposed DTT model, section 4 presents experimental analyses, and section 5 concludes the paper.

2. Related works

2.1. Research on intrusion detection based on Transformer

This section presents a study on NID utilizing Transformer or TCN. Regarding a thorough assessment of Transformer, Manocchio et al. [18] introduced the FlowTransformer framework, which is capable of directly replacing various Transformer components. It lays the foundation for further exploration of Transformer-based approaches in NID. Han et al. [19] merged n-gram frequency with a time-aware Transformer to capture session-level and packet-level features. Despite achieving improved performance compared to previous models, the approach faces hindrances in processing encrypted traffic. Nguten et al. [20] utilized bidirectional encoder representations from Transformers for intrusion detection research. Wu et al. [21] employed a robust Transformer-based IDS to reconstruct feature representations. Nam et al. [22] proposed a bi-directional generative pretrained transformer for attack detection in the Controller Area Network. These Transformer-based models consistently demonstrate superior performance for NID in the supervised domain, providing a solid foundation for our study. In the semi-supervised domain, Li et al. [17] introduced an extreme semi-supervised framework based on Transformer, enhancing performance in the presence of limited labeled data. Its multi-level feature extraction module inspired us to design the FLP coding module.

2.2. Research on intrusion detection based on TCN

Cheng et al. [11] introduced an approach for in-vehicle NID by incorporating global attention into a TCN. It proves that TCN can not only focus on local features but also extract temporal relationships within the context. Sadique et al. [23] utilized TCN for sequential and predictive analysis of heterogeneous threat data, aiming to detect and thwart botnets effectively. Cai et al. [24] proposed a model for detecting malicious network traffic, leveraging bidirectional TCN (BiTCN) and a multi-head self-attention mechanism. This model successfully mitigates data imbalance issues and enhances the accuracy of NID. XGBoost-TCN [25] integrates Extreme Gradient Boosting Decision Tree and TCN for mobile edge computing scenarios, with TCN refining deep timing information within the features. The SSAE-TCN model combines a stacked sparse encoder with TCN within a federated learning framework [26]. It achieves efficient NID and preserves privacy in Internet of Things data.

2.3. Other deep learning models for intrusion detection

Hassan et al. [27] presented a hybrid deep learning model for efficient NID in large-scale data environments. This model integrates a convolutional neural network with a weight-dropped LSTM network, showcasing improved performance. Sheykhkanloo et al. [28] addressed the challenge of insider threat detection in an extremely imbalanced dataset using a spread subsample technique. Xiao et al. [29] proposed Extended Byte-Byte Segmentation Neural Networks (EBSNN) for NID. While it showed effectiveness on their collected dataset, its generalization to other common datasets was poor. In the realm of mobile ad-hoc networks (MANETs), Madhu [30] proposed a model that utilizes COOT optimization and a hybrid LSTM-KNN classifier to bolster network security. Venkateswaran et al. [31] introduced a neuro deep learning wireless IDS tailored for MANETs.

Zipperle et al. and Yang et al. [32, 33] provided comprehensive summaries of recent intrusion detection research. However, there are few studies that organically combine Transformer and TCN techniques in the field of NID. In view of this, our study builds upon Transformer and TCN frameworks. We optimize the multi-head self-attention module in Transformer and introduce a frequency domain module to capture frequency domain features. Based on the above ideas, we propose a novel NID model called DTT.

3. Methodology

Fig. 1 presents the overall architecture of the DTT. We preprocess the input data before feeding it into the DTT for training and classification. The DTT model comprises three components: the input encoding module FLP, the DTT module, and the classification classifier. FLP combines categorical and numerical fields of network data, automatically capturing their advanced features. The DTT module consists of the DFE module and the TCN module. DFE extracts both frequency-domain and time-domain information from the data stream. TCN is used to capture long-term dependency and supplement global features.

The complete flow algorithm of the data stream from the initial stage to the final classification stage is shown in Algorithm 1. Initially, the data undergoes pre-processing, where one-hot encoding and min-max normalization are applied (line 1). This step converts categorical variables into numerical form, enhancing the interpretability and manageability of these features for machine learning models. Following pre-processing, the data is converted into continuous feature vectors using the FLP encoding method (line 3). Subsequently, the processed data is utilized for training with the DTT model (lines 4-6) to classify the attack flow based on header information (line 7).

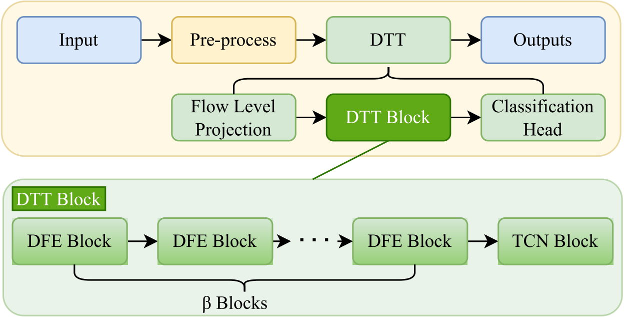

Figure 1. Overall architecture of the DTT model.

```
Algotithm 1: Full Process Algorithm
  Input: dataset X
  Output: Classification results
1 Pre-processing: data → one-hot encoding → min-max normalization by Eq (1);
2 for epoch = 1 to n do
3     FLP: $x' = \mathcal{E}(x^*)$;
4     For $\beta = 1$ to $l$ do
5         $x' = Transformer(x')$;
6     $\tau = TCN(x')$
7     Classification($\tau$) → results;
8 end for
```

The min-max normalization formula is shown in Equation (1).

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}}. \tag{1}$$

In the formula and algorithm, x is the value of any column of the original data set, $x_{\min}$ is the minimum value obtained by counting the whole column, $x_{\max}$ is the maximum value, and x^* is the normalized data value. FLP coding $\mathcal{E}$ is a fully connected layer that maps data onto continuous feature vectors.

3.1. Input encoding options

The function of data pre-processing is to transform the network stream into a format appropriate for model training. Unlike the preprocessing stage, the input encoding constitutes part of the model and integrates into the training phase. The input encoding is not mandatory, which means we can enter the data stream directly into the DTT model for training after the preprocessing stage. Our proposed FLP encoding approach encodes the categorical and numerical fields together. First, the categorical fields are one-hot encoded by the pre-processing part, followed by concatenation with the numerical fields and then normalization. The data is then passed to the FLP layer, which is a fully connected layer that maps the data flow into continuous feature vectors and captures high-dimensional features.

The typical input encodings comprise the Lookup-Based Embedding Layer, Dense Embedding Layer, Linear Projection Layer [34], and Flow Level Embedding [18]. The initial three encode solely categorical fields and omit numerical fields, whereas the final one encodes both categorical fields and numerical values. In particular, there is an additional method called No Encoding, which involves preprocessing the data stream and directly inputting it into the DTT model without encoding the input. This dissertation investigates the impact of these six encodings on the model's performance in Section IV's experimental section.

3.2. DFE block

The Dual-domain Feature Extraction (DFE) block is comprised of two principal branches, as illustrated in Fig. 2. The time domain branch employs an attention mechanism similar to the encoding layer in the Transformer, specifically designed for extracting local time domain features. In contrast, the spectral domain branch employs a Fast Fourier Transform (FFT) layer to substitute the self-attention block in the Transformer, facilitating frequency domain information extraction. This unique design allows our DFE block to simultaneously extract both time- and frequency domain information.

The spectral domain branch of DFE focuses on extracting frequency domain features based on attributes like arrival time, payload length, and protocol type. This is achieved through a spectral gating network with FFT and

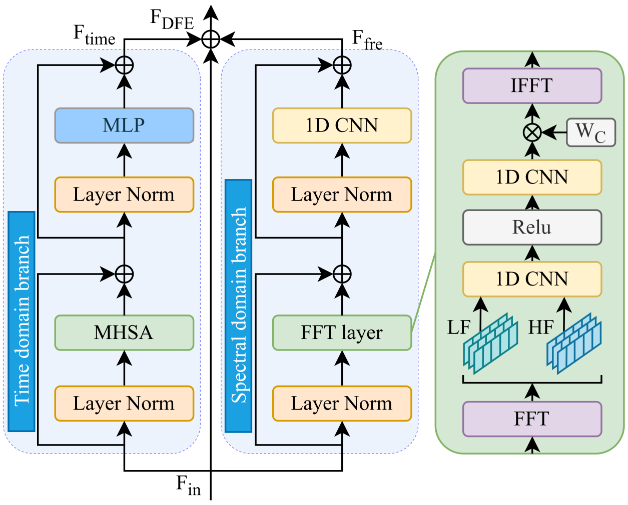

Figure 2. Overall architecture of the DTT model.

1D-CNN layers. Illustrated in Fig. 2, the FFT layer comprises a FFT operation, two convolutional layers, and an IFFT process. The FFT operation transforms data from physical to spectral space, as shown in Equation (2).

$$F_k = \sum_{w=0}^{W-1} F_{in} \cdot e^{-i2\pi\frac{wk}{W}}, (1 \le k \le W), \tag{2}$$

where F_k is a frequency component of F_{in} with the frequency of $2\pi k/W$. The FFT principles indicate that each spectrum frequency is a composite of all time domain points. Thus, frequency domain representation equates to global time series feature extraction. Notably, FFT outputs complex number vectors, which are unsuitable for direct input to deep learning algorithms. To resolve this, real and imaginary parts are channel-wise concatenated, as shown in Equation (3).

$$Z_k = \text{Concatenate}\big(R(F_k), I(F_k)\big). \tag{3}$$

Two 1D convolution neural networks (1d CNNs) are used to extract the frequency domain features, see Equation (4).

$$F(Z_k) = f_{conv}\left(ReLU\big(f_{conv}(Z_k)\big)\right), \tag{4}$$

where f_{conv} representing the 1D CNN. The extracted features are feed in IFFT layer, which reverts frequency domain features back to the time domain through IFFT, see Equation (5)

$$X = IFFT\big(F(Z)\big). \tag{5}$$

Learnable weight parameters are used to emphasize different frequency components, efficiently capturing the data's frequency domain characteristics. These parameters are optimized via back-propagation methods. Post-IFFT, the spectral data undergoes layer normalization and is processed through a 1D-CNN for feature calibration. Ultimately, DFE efficiently extracts and characterizes frequency domain features through a spectral gating network, which fully utilizes the FFT and IFFT operations alongside the learnable weighting parameters.

The time domain branch of DFE focuses on extracting time domain features. It is also utilized to extract local high-frequency details, complementing the global low-frequency semantics captured by the frequency domain branch. This comprises a sequence of normalization and manipulation components, implemented to effectively capture pertinent input data. The attention layer is composed of a normalization layer to begin with, which serves to normalize the input data. Secondly, the model benefits from the Multi-Headed Self-Attention (MHSA) layer, which facilitates the simultaneous consideration of relationships between various input locations for an improved capture of contextual information. The MHSA utilizes a self-attention mechanism founded on a trainable triad (query, key, value). The query and key are employed to calculate the weight score allocated to each value, which is then used to compute the output via a weighted sum of values. This utilizes dot product attention, enabling parallelization of computation and reducing training time. Its calculation Equation (6) is shown below.

$$\mathrm{SelfAttention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right), \tag{6}$$

where Q, K, and V represent the matrices for Query, Key, and Value, respectively, and d_k is the dimension of Key. The formula for MHSA is presented in Equation (7).

$$\begin{cases} Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V i = 1, \dots, n \\ \mathrm{head}_i = \mathrm{Attention}\ (Q_i, K_i, V_i) i = 1, \dots, n \\ \mathrm{MultiHead}\ (Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \dots, \mathrm{end}_n) W^o \end{cases} \tag{7}$$

After conducting MHSA, layer normalization is once again carried out to ensure stability within the layers. Subsequently, it is immediately followed by the MLP module for further feature mapping and extraction.

Finally, the features from each branch of the DFE are integrated with the input features F_{in} (see Equation (8)).

$$F_{DFE} = F_{fre} + F_{time} + F_{in}. \tag{8}$$

3.3. TCN module

The Temporal Convolutional Network (TCN) integrates dilations and residual connections with causal convolutions. Causal convolutions prevent information leakage from the future to the past. Specifically, the output at time *t* undergoes convolution solely with elements from time *t* and earlier in the previous layer. The utilization of dilated convolutions enables an exponentially large receptive field. It allows the model to capture intricate temporal patterns while maintaining computational efficiency. Formally, for a sequence input $x = \{x_0, \dots, x_{t-1}, x_t\} \in \mathbb{R}^n$ and a filter $g: \{0, \dots, k_1\}$, the output of the hidden layer of x_t is defined as Equation (9).

$$h(t) = \sum_{i=0}^{k-1} g(i) \cdot x_{t-d \cdot i}, \tag{9}$$

where *d* is the dilation factor, *k* is the filter size. The dilated convolutions enable the incorporation of information from distant time steps through the parameter d. It enhances TCN's capacity to efficiently capture intricate temporal patterns. Given that the receptive field of TCN relies on the network depth, stabilizing a deeper and larger TCN is critical. The residual connections are proven to enhance the performance of very deep networks. The process is shown in Equation (10).

$$o(\mathbf{x}) = \sigma\big(\mathbf{x} + \mathcal{G}(\mathbf{x})\big), \tag{10}$$

where $\mathcal{G}$ refers to the transformations of **x** and σ denotes the sigmoid activation.

The unique formulation of TCN and its emphasis on parallelization during training distinguish it as a powerful tool in the realm of time series analysis. It offers enhanced performance and scalability in comparison to traditional recurrent architectures.

4. Experiments

4.1. Setup

Experimental configuration

All experiments are currently conducted on a Windows operating system using a computer equipped with an Intel (R) Xeon (R) CPU E5-2678 v3 @ 2.50 GHz and an NVIDIA GeForce RTX 2080 Ti graphics processor. Each experiment is repeated at least three times to ensure result stability. The best-performing results are chosen as the final evaluation metrics from these repeated experiments to avoid any negative influence from poor model initialization.

Datasets

We select the stream-format versions of the NF-CSE-CIC-IDS2018 (NCCI) and NF-UNSW-NB15 (NUB) datasets [35] for evaluating the DTT model.

- The NCCI dataset is a NetFlow-based dataset generated from the original pcap file of CSE-CIC-IDS2018 [36]. The total number of datastreams is 8,392,401, of which 1,019,203 are attack samples and 7,373,198 are benign samples.

- The NUB dataset is the NetFlow format of UNSW-NB15 [37]. The total number of data flows is

2,390,275, with 95,053 being attack samples and 2,295,222 being benign samples. These attacks are further categorized into nine sub-categories.

Data pre-processing

After obtaining the dataset, the next step is data pre-processing. In this paper, we refer to the method of literature [38] in the preprocessing process and divide the dataset into a training dataset and a validation dataset according to a ratio of 90% and 10%. The subsequent stage involves converting the data into a format that can be recognized by the input encoding module, known as the input encoding session.

Various hyperparameter configurations can influence both model convergence speed and experimental outcomes, as detailed in Table 1. When training the DTT model, we use the Adam optimizer to adjust parameters automatically, including the learning rate. However, to avoid the risk of convergence at suboptimal local points, we undertake experiments to establish an appropriate learning rate.

In this experiment, two evaluation metrics are utilized, namely balanced accuracy and F1-score [39]. The metrics are computed from the confusion matrix presented in Table 2. The specific formulas for these metrics are presented in Equations (11) and (12).

$$\begin{cases} S_1 = TP/(TP + FN) \\ S_2 = TN/(TN + FP) \\ \text{balanced accuracy} = (S_1 + S_2)/2 \end{cases} \tag{11}$$

$$\begin{cases} \text{precision} = TP/(TP + FP) \\ \text{recall} = TP/(TP + FN) \\ \text{F1-score} = (\text{precision} + \text{recall})/2 \end{cases} \tag{12}$$

The formula shows that the F1-score indicator is more comprehensive. Therefore, we focus on analysing the F1-score in the experiment.

4.2. Results and Discussion

The experiments in the study are organized into four main sections. Firstly, we examine the impact of different input encoding methods on model performance. Subsequently, we focus on the role of each component within the DFE block. Thirdly, we explore how adjusting the number of layers (represented as β) of the DFE block affects outcomes. This allows us to evaluate the performance of DTT across varying complexity levels and scrutinize the influence of selected layers on model effectiveness. Finally, we compare our proposed model with recent intrusion detection models, assessing its advantages and competitiveness within the same experimental framework and dataset.

Input encodings

In this section, we investigate the effect of different input encoding methods on model performance. For this experiment, we set the β value of the DFE block to 2 and the number of Attention Heads to 2. This configuration indicates that the DTT Block comprises two DFE blocks, with each Attention Block within the DFE block containing two Attention Heads. This represents the optimal model configuration obtained from our experiments, which will be discussed further in subsequent sections. Fig. 3 illustrates the results of training the DTT model using different input encoding methods on the NCCI and NUB datasets. It is evident that there are no significant differences in F1-score across different input coding methods. This suggests that these methods have a limited impact on the final performance of DTT. Nevertheless, FLP demonstrates a slight advantage in terms of F1-score. Detailed experiment specifications are provided in Table 3, offering additional perspectives for model evaluation. As depicted in the table, despite the little impact of various encoding methods on model performance, FLP has the fewest number of parameters. It indicates its superior efficiency in model operation.

Table 1. Parameter configuration.

Parameter name	Value
Batch size	64
Dropout	0.5
TCN_dilation_factor	2^m
TCN_kernel_size	3,4,5
Attention heads	2
Learning rate	0.001
β	1,2,3

Table 2. Confusion matrix.

The real situation	Projection result	
	Positive	Negative
True	TP	FP
False	FP	TN

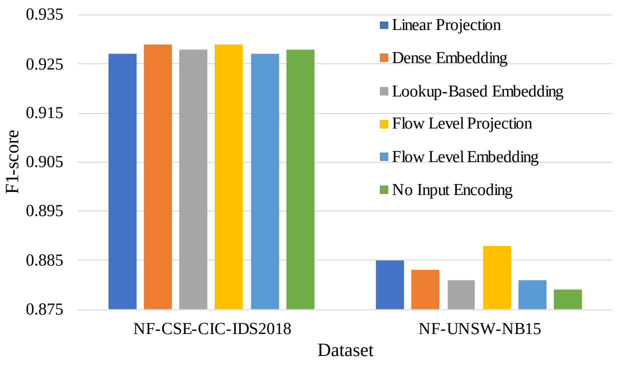

Figure 3. Results for different input encoding under DTT model (β = 2, Attention Heads = 2).

Table 3. Effect of different input encoding on the model.

Dataset	Input encoding method	Parameters	F1-score	Balanced accuracy
NCCI	Linear projection	411345	0.927	0.936
	Dense Embedding	412075	0.929	0.934
	Lookup-Based Embedding	412075	0.928	0.936
	Flow Level Projection	110259	0.929	0.935
	Flow Level Embedding	130245	0.927	0.936
	No Input Encoding	723553	0.928	0.935
NUB	Linear projection	412449	0.885	0.980
	Dense Embedding	413893	0.883	0.975
	Lookup-Based Embedding	413893	0.881	0.981
	Flow Level Projection	118581	0.888	0.982
	Flow Level Embedding	139863	0.881	0.981
	No Input Encoding	682163	0.879	0.982

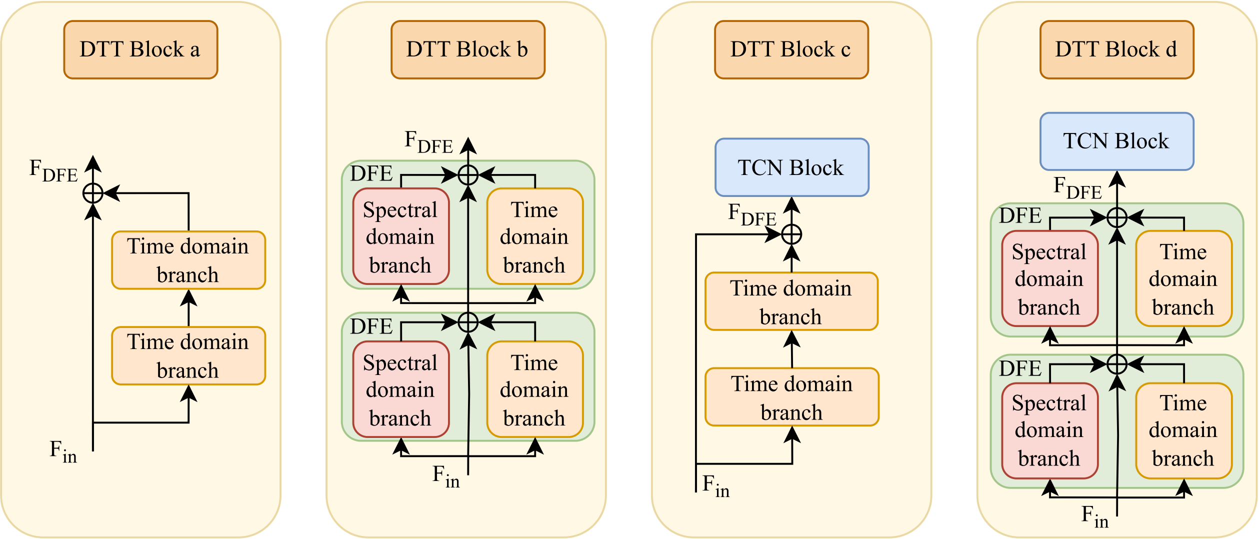

Figure 4. Structure of DTT for each group

Ablation experiments

As demonstrated in Fig. 1, the DTT module comprises the DFE and TCN blocks. The attention mechanism in the DFE block has been extensively showcased in recent studies. This experiment aims to examine the impact of the spectral domain branch in the DFE block and TCN on model performance. By assessing the outcomes of various combinations of these elements, we can develop a comprehensive understanding of the overall performance of the DTT model.

The experiment was divided into four distinct groups labeled a, b, c, and d. In Group a, the DTT module comprised solely two layers of time domain branches in DFE. Group b's module included two DFE blocks, each containing both time and spectral domain branches. The structure of Group c's module involved two layers of time domain branches and a TCN block. Group d's experiments utilized the complete DTT module, consisting of two DFE blocks and a TCN block. The structure of the four model groups is displayed in Fig. 4 and Table 4. For all experimental models, FLP encoding is chosen as the input encoding, and the parameter β is illustrated in the figure. The other hyperparameters remained constant.

The experiment results presented in Fig. 5 and Table 5 reveal distinct performance differences among the four groups. Group b outperforms Group a across both datasets, indicating the effectiveness of incorporating spectral domain information within the DFE block. Similarly, Group c shows considerable performance improvement over Group a, highlighting the effectiveness of using the

TCN block to capture long-term dependencies. Notably, Group d exhibits the highest model performance among all groups. This suggests that the integration of DFE and TCN blocks yields the most substantial enhancement in model performance.

Table 4. Configuration of ablation experiment groups

	DFE with Spectral domain branch	TCN
Group a	×	×
Group b	√	×
Group c	×	√
Group d	√	√

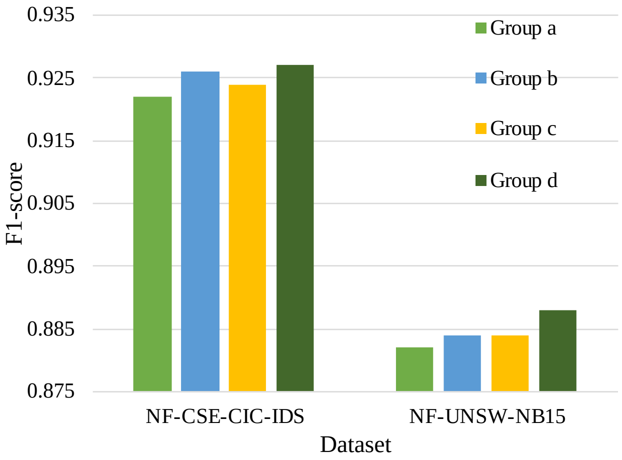

Figure 5. Diagram of the ablation experiment with the DTT module. β = 2, Attention Heads = 2).

Table 5. Results of ablation experiments with DTT module (F1-score/ Balanced Accuracy)

Datasets	NCCI	NUB
Group a	0.922/0.934	0.881/0.980
Group b	0.926/0.935	0.882/0.982
Group c	0.924/0.936	0.882/0.981
Group d	0.927/0.936	0.888/0.981

Performance comparison of the DFE block with varying numbers of layers (with differing β)

This experiment aims to examine how the number of layers of the DFE block influences the overall effectiveness of the model. Understanding how various layers affect performance helps us achieve optimal model design and selection. Fig. 6 displays the model diagram with varying numbers of layers.

The results of the experiment are displayed in Table 6. It reveals that the DTT model performs more effectively when β is set to 2 on the NUB dataset. In addition, on the NCCI dataset, the DTT model with β set to 2 shows comparable performance to the one with β set to 3, but the former entails fewer parameters. This implies that it consumes less memory and operates more efficiently. Consequently, it can be concluded that the DTT model with two DFE block layers outperforms other configurations in terms of performance.

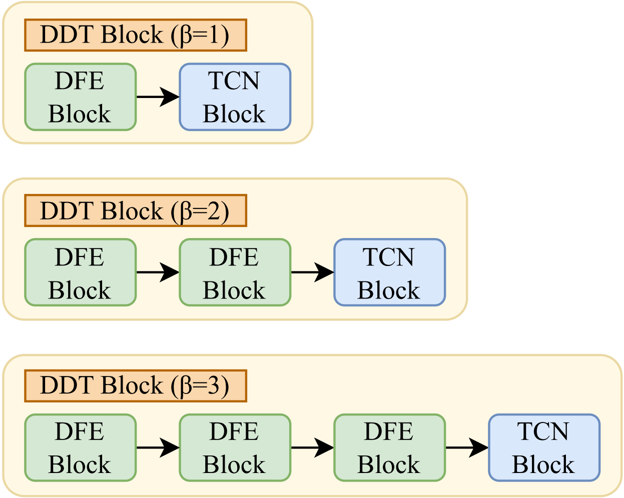

Figure 6. DTT model corresponding to different number of layers.

Table 6. Impact of Various β on Model Performance

Datasets	β	F1-score	Balanced Accuracy
	1	0.895	0.902
NCCI	2	0.927	0.936
	3	0.926	0.937
	1	0.876	0.980
NUB	2	0.888	0.981
	3	0.881	0.979

Research comparison experiments

In this section, the DTT model proposed in this research paper is compared with other recent intrusion detection models. To ensure fairness, the comparison experiments are conducted under the same conditions, including identical datasets, preprocessing methods, and input encoding methods. Details of the comparative outcomes are presented in Table 7. The DTT model exhibits improvements in F1-score ranging from 0.6 to 6.8 on the NCCI dataset and from 0.4 to 3.5 on the NUB dataset compared to other models. These findings indicate that our model surpasses other models in recent years under the same experimental conditions. Some studies excel in the preprocessing of the data or other aspects rather than the model itself. This experiment only demonstrates that the DTT model itself outperforms the models investigated in these studies.

Table 7. Impact of Various β on Model Performance.

Dataset	Reference	Model	Balanced Accuracy	F1-score
NCCI	Liam et al. [18]	transformer	0.923	0.914
	Cai et al. [24]	BiTCN	0.925	0.921
	Yang et al. [26]}	SSAE-TCN	0.911	0.859
	Hassan et al. [27]}	hybrid DL	0.923	0.895
	Jiao et al. [25]	XGBoost-TCN	0.921	0.892
	Xiao et al. [29]	EBSNN	0.917	0.874
	Madhu et al. [30]	LSTMKNN	0.922	0.897
	This Paper	DTT	0.936	0.927
NUB	Liam et al. [18]	transformer	0.945	0.853
	Cai et al. [24]	BiTCN	0.962	0.878
	Yang et al. [26]}	SSAE-TCN	0.954	0.867
	Hassan et al. [27]}	hybrid DL	0.967	0.879
	Jiao et al. [25]	XGBoost-TCN	0.974	0.88
	Xiao et al. [29]	EBSNN	0.979	0.884
	Madhu et al. [30]	LSTMKNN	0.972	0.877
	This Paper	DTT	0.981	0.888

5. Conclusion

In this study, we introduced a Dual-domain intrusion detection model based on TCN and Transformer, named DTT. It integrates a frequency domain module with Transformer and TCN. We also utilized an efficient input encoding method tailored to the attention heads of Transformer. The Transformer with the spectral domain branch comprehensively extracts frequency and time domain information. This, coupled with TCN's capability to capture long-term dependencies, enhances the accuracy and efficiency of DTT for NID. As a result, DTT contributes to safeguarding network security and preventing system breakdowns. Extensive experiments conducted on two large-scale intrusion detection datasets demonstrated the performance of DTT, suggesting its potential to advance the field of NID. However, it is worth noting that this approach requires extensive pre-training, which may not be suitable for real-time intrusion detection scenarios. Additionally, the black-box nature of deep learning models poses challenges in gaining trust in network security management. In future work, we aim to optimize the model's real-time processing capabilities and explore interpretable methods for NID.

References

[1] Patil, D.R. and Pattewar, T.M. (2022) Majority voting and feature selection based network intrusion detection system. *EAI Endorsed Transactions on Scalable Information Systems* **9**(6): e6. doi: 10.4108/eai.4-4-2022.173780, https://publications.eai.eu/index.php/sis/article/view/350.

[2] Novaes Neto, N., Madnick, S., de Paula, M.G., Malara Borges, N. *et al.* (2021) A case study of the capital one data breach: Why didn't compliance requirements help prevent it? *Journal of Information System Security* **17**(1): 49–78. http://dx.doi.org/10.2139/ssrn.3542567.

[3] Peisert, S., Schneier, B., Okhravi, H., Massacci, F., Benzel, T., Landwehr, C., Mannan, M. *et al.* (2021) Perspectives on the solarwinds incident. *IEEE Security & Privacy* **19**(2): 7–13. doi: 10.1109/MSEC.2021.3051235.

[4] Yin, J., Tang, M., Cao, J., You, M., Wang, H. and Alazab, M. (2023) Knowledge-driven cybersecurity intelligence: Software vulnerability coexploitation behavior discovery. *IEEE Transactions on Industrial Informatics* **19**(4): 5593–5601. doi: 10.1109/TII.2022.3192027.

[5] Zhang, C., Costa-Perez, X. and Patras, P. (2022) Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. *IEEE/ACM Transactions on Networking* **30**(3): 1294–1311. doi: 10.1109/TNET.2021.3137084.

[6] Blaise, A., Bouet, M., Conan, V. and Secci, S. (2020) Detection of zero-day attacks: An unsupervised port-based appr

oach. *Computer Networks* **180**: 107391. doi: https://doi.org/10.1016/j.comnet.2020.107391, https://www.sciencedirect.com/science/article/pii/S1389128620300761.

[7] Liu, F., Zhou, X., Cao, J., Wang, Z., Wang, T., Wang, H. and Zhang, Y. (2022) Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional lstm-cnn. *IEEE Transactions on Knowledge and Data Engineering* **34**(6): 2626–2640. doi: 10.1109/TKDE.2020.3014806.

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u. *et al.* (2017) Attention is all you need. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R. [eds.] *Advances in Neural Information Processing Systems* (Curran Associates, Inc.), **30**. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[9] Abdel-Basset, M., Hawash, H., Chakrabortty, R.K. and Ryan, M.J. (2021) Semi-supervised spatiotemporal deep learning for intrusions detection in iot networks. *IEEE Internet of Things Journal* **8**(15): 12251–12265. doi: 10.1109/JIOT.2021.3060878.

[10] Liang, P., Yang, L., Xiong, Z., Zhang, X. and Liu, G. (2024) Multi-level intrusion detection based on transformer and wavelet transform for iot data security. *IEEE Internet of Things Journal* : 1–1doi: 10.1109/JIOT.2024.3369034.

[11] Cheng, P., Xu, K., Li, S. and Han, M. (2022) Tcan-ids: Intrusion detection system for internet of vehicle using temporal convolutional attention network. *Symmetry* **14**(2). doi: 10.3390/sym14020310, https://www.mdpi.com/2073-8994/14/2/310.

[12] Shao, M., Qiao, Y., Meng, D. and Zuo, W. (2023) Uncertainty-guided hierarchical frequency domain transformer for image restoration. *Knowledge-Based Systems* **263**: 110306. doi: https://doi.org/10.1016/j.knosys.2023.110306, https://www.sciencedirect.com/science/article/pii/S0950705123000564.

[13] Alvi, A.M., Siuly, S. and Wang, H. (2023) A long short-term memory based framework for early detection of mild cognitive impairment from eeg signals. *IEEE Transactions on Emerging Topics in Computational Intelligence* **7**(2): 375–388. doi: 10.1109/TETCI.2022.3186180.

[14] Fu, C., Li, Q., Shen, M. and Xu, K. (2023) Frequency domain feature based robust malicious traffic detection. *IEEE/ACM Transactions on Networking* **31**(1): 452–467. doi: 10.1109/TNET.2022.3195871.

[15] Zhong, Z., Sun, L., Subramani, S., Peng, D. and Wang, Y. (2023) Time series classification for portable medical devices. *EAI Endorsed Transactions on Scalable Information Systems* **10**(4): e19. doi: 10.4108/eetsis.v10i3.3219, https://publications.eai.eu/index.php/sis/article/view/3219.

[16] Singh, R., Subramani, S., Du, J., Zhang, Y., Wang, H., Miao, Y. and Ahmed, K. (2023) Antisocial behavior identification from twitter feeds using traditional machine learning algorithms and deep learning. *EAI Endorsed Transactions on Scalable Information Systems* **10**(4): e17. doi: 10.4108/eetsis.v10i3.3184, https://publications.eai.eu/index.php/sis/article/view/3184.

[17] Li, Y., Yuan, X. and Li, W. (2022) An extreme semi-supervised framework based on transformer for network intrusion detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM ’22 (New York, NY, USA: Association for Computing Machinery): 4204–4208. doi: 10.1145/3511808.3557549, https://doi.org/10.1145/3511808.3557549.

[18] Manocchio, L.D., Layeghy, S., Lo, W.W., Kulatilleke, G.K., Sarhan, M. and Portmann, M. (2024) Flowtransformer: A transformer framework for flow-based network intrusion detection systems. *Expert Systems with Applications* **241**: 122564. doi: https://doi.org/10.1016/j.eswa.2023.122564, https://www.sciencedirect.com/science/article/pii/S095741742303066X.

[19] Han, X., Cui, S., Liu, S., Zhang, C., Jiang, B. and Lu, Z. (2023) Network intrusion detection based on n-gram frequency and time-aware transformer. *Computers & Security* **128**: 103171. doi: https://doi.org/10.1016/j.cose.2023.103171, https://www.sciencedirect.com/science/article/pii/S0167404823000810.

[20] Nguyen, L.G. and Watabe, K. (2022) Flow-based network intrusion detection based on bert masked language model. In *Proceedings of the 3rd International CoNEXT Student Workshop*, CoNEXT-SW ’22 (New York, NY, USA: Association for Computing Machinery): 7–8. doi: 10.1145/3565477.3569152, https://doi.org/10.1145/3565477.3569152.

[21] Wu, Z., Zhang, H., Wang, P. and Sun, Z. (2022) Rtids: A robust transformer-based approach for intrusion detection system. *IEEE Access* **10**: 64375–64387. doi: 10.1109/ACCESS.2022.3182333.

[22] Nam, M., Park, S. and Kim, D.S. (2021) Intrusion detection method using bi-directional gpt for in-vehicle controller area networks. *IEEE Access* **9**: 124931–124944. doi: 10.1109/ACCESS.2021.3110524.

[23] Sadique, F. and Sengupta, S. (2022) Modeling and analyzing attacker behavior in iot botnet using temporal convolution network (tcn). *Computers & Security* **117**: 102714. doi: https://doi.org/10.1016/j.cose.2022.102714, https://www.sciencedirect.com/science/article/pii/S0167404822001092.

[24] Cai, S., Xu, H., Liu, M., Chen, Z. and Zhang, G. (2024) A malicious network traffic detection model based on bidirectional temporal convolutional network with multi-head self-attention mechanism. *Computers & Security* **136**: 103580. doi: https://doi.org/10.1016/j.cose.2023.103580, https://www.sciencedirect.com/science/article/pii/S016740482300490X.

[25] Jiao, X., Li, J. and Wen, M. (2022) Intrusion detection based on feature selection and temporal convolutional network in mobile edge computing environment. *International Journal of Network Security* **24**(2): 286–295. doi: 10.6633/IJNS.202203_24(2).11.

[26] Yang, R., He, H., Xu, Y., Xin, B., Wang, Y., Qu, Y. and Zhang, W. (2023) Efficient intrusion detection toward iot networks using cloud-edge collaboration. *Computer Networks* **228**: 109724. doi: https://doi.org/10.1016/j.comnet.2023.109724, https://www.sciencedirect.com/science/article/pii/S138912862300169X.

[27] Hassan, M.M., Gumaei, A., Alsanad, A., Alrubaian, M. and Fortino, G. (2020) A hybrid deep learning model for efficient intrusion detection in big data environment. *Information Sciences* **513**: 386–396. doi: https://doi.org/10.1016/j.ins.2019.10.069, https://www.sciencedirect.com/science/article/pii/S0020025519310382.

[28] Sheykhkanloo, N.M. and Hall, A. (2020) Insider threat detection using supervised machine learning algorithms on an extremely imbalanced dataset. *Int. J. Cyber Warf. Terror.* **10**(2): 1–26. doi: 10.4018/IJCWT.2020040101, https://doi.org/10.4018/IJCWT.2020040101.

[29] Xiao, X., Xiao, W., Li, R., Luo, X., Zheng, H. and Xia, S. (2022) Ebsnn: Extended byte segment neural network for network traffic classification. *IEEE Transactions on Dependable and Secure Computing* **19**(5): 3521–3538. doi: 10.1109/TDSC.2021.3101311.

[30] G., M. (2022) Design of intrusion detection and prevention model using coot optimization and hybrid lstm-knn classifier for manet. *EAI Endorsed Transactions on Scalable Information Systems* **10**(3): e2. doi: 10.4108/eetsis.v10i3.2574, https://publications.eai.eu/index.php/sis/article/view/2574.

[31] Venkateswaran, N. and Prabaharan, S.P. (2022) An efficient neuro deep learning intrusion detection system for mobile adhoc networks. *EAI Endorsed Transactions on Scalable Information Systems* **9**(6): e7. doi: 10.4108/eai.4-4-2022.173781, https://publications.eai.eu/index.php/sis/article/view/351.

[32] Zipperle, M., Gottwalt, F., Chang, E. and Dillon, T. (2022) Provenance-based intrusion detection systems: A survey. *ACM Comput. Surv.* **55**(7). doi: 10.1145/3539605, https://doi.org/10.1145/3539605.

[33] Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y. and Han, H. (2022) A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security* **116**: 102675. doi: https://doi.org/10.1016/j.cose.2022.102675, https://www.sciencedirect.com/science/article/pii/S0167404822000736.

[34] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013) Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13 (Red Hook, NY, USA: Curran Associates Inc.): 3111–3119.

[35] Sarhan, M., Layeghy, S. and Portmann, M. (2022) Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks &amp; Applications* **27**(1): 357 – 370. https://search.ebscohost.com/login.aspx?direct=true&amp;db=asn&amp;AN=155954870&amp;lang=zh-cn&amp;site=eds-live.

[36] Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A. (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Mori, P., Furnell, S. and Camp, O. [eds.] *Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Madeira - Portugal, January 22-24, 2018* (SciTePress): 108–116. doi: 10.5220/0006639801080116, https://doi.org/10.5220/0006639801080116.

[37] 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. doi: 10.1109/MilCIS.2015.7348942.

[38] Gao, J. and Bhardwaj, A. (2022) Network intrusion detection method combining cnn and bilstm in cloud computing environment. *Intell. Neuroscience* **2022**. doi: 10.1155/2022/7272479, https://doi.org/10.1155/2022/7272479.

[39] Sun, L., Li, C., Liu, B. and Zhang, Y. (2023) Class-driven graph attention network for multi-label time series classification in mobile health digital twins. *IEEE Journal on Selected Areas in Communications* **41**(10): 3267–3278. doi: 10.1109/JSAC.2023.3310064.