IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 June 2024, accepted 13 July 2024, date of publication 16 July 2024, date of current version 24 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3429420

RESEARCH ARTICLE

TSSAN: Time-Space Separable Attention Network for Intrusion Detection

RUI XU, QI ZHANG, AND YUNJIE ZHANG
School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215008, China
Corresponding author: Rui Xu (rxu1026@stu.suda.edu.cn)

ABSTRACT With the continuous evolution of novel network attacks, traditional Intrusion Detection Systems (IDSs) have commonly employed Deep Neural Networks (DNNs) for intrusion detection. However, the effectiveness of a DNN in this respect is closely related to the quality of the training data set, and large-scale network traffic data are difficult to label accurately. Therefore, some challenges still need to be addressed to detect network attacks. In this paper, we introduce a Time-Space Separable Attention Network (TSSAN) for intrusion detection. TSSAN utilizes depth wise separable convolution and a time-space self-attention mechanism to effectively extract temporal and spatial features. By extracting the common features from the unlabeled data, TSSAN significantly enhanced the detection performance for rare attack types. Experimental evaluations were conducted using UNSW-NB15 and CICIDS-2017 datasets. Meticulous experiments for evaluating the individual components of the model were rigorously carried out using the CICIDS-2017 dataset. In the unsupervised learning experiment, our method achieved 0.86 and 0.92 f1score in the two datasets. In semi-supervised learning, the experiment showed that our method performed significantly better than the traditional deep learning method when the labelled data were gradually reduced.

INDEX TERMS Intrusion detection, deep learning, network security, self attention, multi-class classification.

I. INTRODUCTION

With the proliferation of network devices and the exponential surge in network traffic, the threat of network attacks is escalating at an alarming rate. The proliferation of novel network security threats further amplifies the suddenness and destructiveness of network security threats. Traditional network intrusion detection systems (IDS) rely on predefined rules or known signatures of attacks, thereby limiting their ability to detect known attacks. As the number of new attack types expands, the detection efficacy of traditional IDS gradually declines, making it challenging to address the growing number of novel network security threats [1].

Traditional techniques such as the encryption-decryption method, protocol control, firewalls, and anti-virus software models have many limitations. Although these methods can successfully identify certain types of attacks, they are ineffective in dealing with a large number of attacks and denial of service (DoS) attacks. They also have low detection rates and high false-alarm rates. Consequently, modern research is increasingly turning to machine learning (ML) techniques for intrusion detection, as they have been shown to outperform traditional methods in terms of recognition rate and efficiency in handling large-scale attacks. Specific ML methods that have been used for intrusion detection include the support vector machine (SVM) [2], K-nearest neighbor (KNN) [3], random forest [4], k-means clustering, and logistic regression. However, traditional ML algorithms have significant limitations when dealing with attacks that have highly integrated features, and they perform poorly when dealing with noisy and multidimensional traffic data [5].

In recent years, deep learning has attracted significant attention owing to its impressive performance in computer vision (CV) [6] and natural language processing (NLP) [7]. Consequently, many researchers have shifted their focus toward exploring the potential of deep learning. Intrusion detection methods based on deep learning can be divided into three stages: data acquisition, feature extraction, and anomaly discrimination [8]. Recent studies have demonstrated its effectiveness [9]. However, most existing studies directly

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan.

utilize language models, such as long short-term memory (LSTM) [10] for feature extraction. These methods typically adopt an end-to-end learning approach to classification. However, such approaches often struggle to effectively capture long-term temporal features and depend heavily on manually labelled data. Consequently, the feature-extraction process is incomplete.

The strategy of employing unsupervised learning to train large-scale model architectures and subsequently fine-tune them on a small labelled dataset has garnered considerable attention [11]. This trend has led to the development of impressive works such as GPT [12], and BERT [13]. Consequently, large unsupervised learning models have become the focus of research.

Moreover, selecting the correct features and dealing with unbalanced data are critical challenges. Researchers often employ the PSO algorithm to automatically select the optimal feature subset and utilize the focus loss function to address the issue of imbalanced categories within a dataset [14]. By optimizing the feature selection process, the data dimension can be decreased and the computational efficiency and generalization ability of the model can be enhanced. The focus loss function enhances the detection ability of minority categories by providing them with greater weight, thereby enabling the model to focus on an accurate classification.

Furthermore, additional factors must be considered for practical application [15]. Industrial Internet of Things (IIOT) data typically exhibit temporal and dynamic qualities, necessitating real-time response and the ability to adapt to evolving data. Given the intricacy of the IIoT environment and the constraints of computing resources, deep anomaly detection models should have a moderate level of complexity to deliver superior detection performance while also being sufficiently efficient to operate on devices with limited resources.

Although there are notable unsupervised learning approaches for anomaly detection in one-dimensional time-series data, the analysis of network traffic data presents distinct challenges. First, network traffic data contains categorical features such as IP addresses, nodes, states and protocols, rather than continuous data. Second, the data size exhibits significant fluctuations [16]. Furthermore, the internal features of the network traffic data exhibited a low level of correlation. The simplistic treatment of network traffic data as one-dimensional time-series data fails to effectively aggregate and extract their features, resulting in suboptimal anomaly detection performance.

In general, many currently available intrusion detection techniques rely significantly on labelled data, which means that inaccurate labelling or the absence of labels in a dataset can significantly affect the performance of the model. Consequently, these models may not be able to handle the intricacy and variability of real-world data effectively, leading to suboptimal results. Furthermore, internal feature processing of network traffic data plays a crucial role in determining the performance of the model. The inability to properly aggregate and extract features can also make it difficult to accurately capture temporal and dynamic characteristics, which are essential for real-time response and adaptation to constantly changing data.

The primary research goal of this study is to propose an intrusion detection model based on unsupervised learning that does not rely on large, manually labelled datasets. This model effectively captures the temporal and spatial characteristics of network traffic and achieves superior detection performance.

The main contributions of this paper are summarized as follows:

- An unsupervised learning model called the Time-Space Separable Attention Network(TSSAN), which incorporates depth wise separable convolution and time-space self-attention mechanisms is proposed. This algorithm demonstrated its ability to accurately and efficiently classify and detect intrusion traffic in the CICIDS2017 and UNSW-NB15 datasets.
- The model was enhanced through the application of Gaussian jitter and Gaussian noise, which have been demonstrated to significantly improve its generalization capacity and decrease the likelihood of overfitting in experimental results.
- The TSSAN demonstrated significant efficiency gains when comparing the fully supervised models to the pre-trained and fine-tuned models. The model achieved faster training and detection times, while simultaneously maintaining minimal loss in performance.
- The model performed well in the small sample scenarios. By leveraging unsupervised learning to extract temporal and spatial features, the model achieves efficient feature learning and significantly improves detection performance with limited labelled data.

The remainder of this paper is organized as follows. In Section II, an introduction to the related work is provided. Section III analyzes the data preprocessing methods and presents the model architecture. Section IV reports the experimental results of the study, including unsupervised learning and fine-tuning experiments conducted on a small labelled dataset. Finally, Section V concludes the paper and offers prospects for future work.

II. RELATED WORKS

A. NETWORK INTRUSION DETECTION

Given the remarkable advancements in deep learning in CV [17], [18] and NLP [19], numerous researchers have endeavored to extend the application of deep learning techniques to network intrusion detection. Research has shown that machine-learning and deep-learning techniques can effectively detect network anomalies [20]. Supervised learning methods often rely on models trained on labelled datasets [21]. Sinha and Manollas [10] proposed a fusion of 1D convolutional neural networks (CNN) and bidirectional

LSTM to capture the temporal features of network traffic data. Similarly, Singh et al. [22] proposed a lightweight network that combines LSTM and Gated Recurrent Units (GRU) to reduce the computational cost. A Self Attention-based Long Short Term Memory (SALSTM) network was employed to evaluate the attack detection capabilities of the proposed framework, with the explainability of the AI-based IDS achieved using the SHapley Additive exPlanations (SHAP) tool [23]. Javeed et al. addressed the security challenges in the Industrial Internet of Things(IIoT) [24] by combining the capabilities of two advanced deep learning (DL) classifiers. Inspired by graph neural networks, several studies have made advancements such as GraphSAGE [25] and GAT [26]. In these studies, IP addresses and ports were mapped to nodes, whereas network connections were represented as edges. These approaches exhibit promising results for the UNSW-NB15 dataset. Considering the computational cost, Corin et al. [27] built a binary classification CNN specifically for Distributed Denial of Service (DDOS) attacks, achieving high-accuracy detection with low computational overhead. However, these approaches rely heavily on manually labelled datasets and are highly dependent on their quality. Despite their superior performance in classification tasks compared with traditional methods, supervised learning methods face challenges when detecting rare anomalies or unknown attacks. Additionally, the adjustment of these models requires extensive costs. Furthermore, the presence of mislabelled data leads to a significant decrease in model accuracy [28].

Owing to the scarcity of labelled data, researchers have increasingly directed their attention toward unsupervised learning approaches. Some researchers have chosen a Generative Adversarial Network(GAN) as the main model for anomaly detection [29], [30]. Cui et al. [31] successfully integrated a clustering algorithm based on the Gaussian Mixture Model (GMM) and Wasserstein Generative Adversarial Network (WGAN) to address the issues of data imbalance and inadequate rare attack samples. Zhong et al. [32] proposed an unsupervised learning intrusion traffic classification model that utilises the Wasserstein divergence target generative adversarial network (WGAN div) and information maximization generative adversarial network (Info GAN) to address the problem of low accuracy in small-sample classification. To reduce latency and enhance efficiency, some researchers have suggested a novel GAN-based IDS that employs temporary convolutional networks (TCNS) and self-attention to detect network attacks. This method was demonstrated to be more precise and quicker than conventional LSTM-based IDS [33]. Although Generative Adversarial Networks (GANs) are effective in detecting anomalies, their use is limited. One issue is the difficulty in training GANs to converge, which can result in pattern collapse. Additionally, the complex distribution of the dataset can lead to deviations in the generated data points from mainstream features, making it difficult for the generative model to effectively restore normal samples [34].

Some researchers used autoencoders for detection. Autoencoders have been widely studied owing to their ease of execution. Catillo et al. [35] proposed a semi-supervised autoencoder-based intrusion detection method that emphasizes its usability and reliability in practical applications. Zhang et al. [36] achieved linear and nonlinear dimensionality reductions using Pearson correlation coefficients and stacked sparse autoencoders, preserving the important features of the original data while reducing redundancy. Lopes et al. [37] trained deep autoencoders to learn compressed representations and utilized low-dimensional representation data to train Deep Neural Network(DNN) classifiers, thereby effectively reducing the need for annotated datasets. Overall, numerous AE variants can be applied to anomaly detection, making this method highly adaptable. However, if outliers are present in the training data, the model learns information about them, resulting in biased learning [34].

B. TIME-SERIES ANOMALY DETECTION

In recent years, significant attention has been paid to processing of time-series data. Time-series data represent a sequence of observations arranged in chronological order. Li and Jung [38] classified anomalies in time-series data into three categories: time points, time intervals, and time series. They conducted a comprehensive review of the latest deep learning techniques used for time-series anomaly detection. The limitations of each method were analyzed, along with the challenges and issues associated with applying deep learning methods to anomaly detection. Alahamade et al. [39] detected anomalies in time-series data by clustering. Inspired by the outstanding performance of CNNs in computer vision, researchers have also begun to apply CNNs to anomaly detection [40]. Dutt et al. [41] utilized a combination of 1-D CNN blocks and conditional random fields (CRFs) to classify sleep labels. Choi et al. [42] transformed a multidimensional time series within each time step into distance images and employed a generative adversarial network (GAN) for anomaly detection based on these images. Adiban et al. [43] optimized the discriminator to utilize GAN in time series data. Moreover, with the exceptional performance of transformer in NLP [44] and CV [45], self-attention mechanisms have also been introduced in anomaly detection. Song et al. [46] employed a transformer encoder for time-series anomaly detection. Wu et al. [47] proposed improvements to the self-attention mechanism, capturing both prior and series associations to emphasize local and global information. Through their combined approach, they achieved a high detection performance. To reduce the computational complexity of Transformer for long sequences, Kitaev et al. [48] replaced the dot-product self-attention mechanism with locality-sensitive hashing. Similarly, Zhou et al. [49] employed distillation techniques to eliminate redundant sparse self-attention, thereby enabling the continuous extraction of salient features. Wu et al. [50] applied a fast Fourier transform to extract

periodic information from one-dimensional time-series data, followed by folding based on different periods and projecting the data into a two-dimensional space. With the rise of contrastive learning, this unsupervised learning approach has also been applied to anomaly detection in time series. Eldele et al. [51] performed data augmentation on time series using weak augmentation (jitter-and-scale) and strong augmentation (permutation-and-jitter), followed by a contrastive learning module for prediction tasks. Yue et al. [52] proposed improvements in the sampling of positive and negative samples, enabling the learning of context representations for arbitrary subsequences at different semantic levels.

The extensive utilization of self-attention mechanisms in time-series analysis has provided compelling evidence of their efficacy in extracting temporal information from long sequences. Their ability to capture long-term dependencies, consider the global context, and adaptively assign attention weights to relevant features makes them powerful tools for feature extraction in time-series anomaly detection and other related applications.

C. TIME-SPACE SELF-ATTENTION

Transformer [44] has demonstrated powerful effectiveness in NLP by utilizing self-attention mechanisms to capture the correlations between different positions within a sequence, thus demonstrating its remarkable feature extraction capabilities. In addition, the Vision Transformer [45] has proven the significant potential of self-attention mechanisms in the field of computer vision. However, the self-attention mechanism applied to both the language and image domains primarily focuses on capturing spatial feature correlations and overlooks the extraction of temporal features.

To overcome this limitation, the time-space self-attention mechanism divides the self-attention mechanism into temporal and spatial self-attention layers. The temporal self-attention layers are designed to capture features between different time steps within sequential data, while the spatial self-attention layers capture correlations among different positions in the sequence. By integrating the advantages of temporal and spatial attention, the time-space self-attention mechanism demonstrates remarkable efficacy in dealing with time-series data that possess intricate spatiotemporal relationships. As a result, it significantly enhances the feature extraction capacity for time-series data [53].

III. METHOD

Network traffic data, although belonging to the category of time-series data, exhibit distinct characteristics compared with common time-series data. The conventional approach involves preprocessing network traffic data and utilizing language models such as RNN and LSTM to extract temporal features. However, this approach relies heavily on labelled data. Mislabelled data can significantly impact the performance of deep learning models. To address these challenges, a model based on Time-Space self-attention for intrusion detection is proposed in this paper.

A. DATASET DESCRIPTION

The UNSW-NB15 dataset was developed by creating a synthetic environment at the UNSW cybersecurity lab using the IXIA tool. This tool allows for the generation of modern, representative network traffic for both normal and abnormal situations in a synthetic environment. UNSW-NB15 represents nine major types of attacks using the IXIA PerfectStorm tool and includes 49 features developed with Argus and Bro-IDS tools, as well as 12 algorithms covering packet characteristics. In contrast to benchmark datasets such as KDD98, KDDCUP99, and NSLKDD, which have limited attack and outdated packet information, UNSW-NB15 provides a more comprehensive set of data [54]. Table 1 presents a overview of features of UNSW-NB15 datasets, providing a clear and concise overview of each feature's type.

The CICIDS2017 dataset [55] encompasses both benign and contemporary common attacks that closely approximate genuine real-world data (PCAPs). This dataset includes the findings of network traffic analysis using CICFlowMeter, which is accompanied by labelled flows categorized by timestamp, source and destination IPs, source and destination ports, protocols, and attacks (CSV files). Additionally, this dataset comprises a comprehensive definition of the extracted features.

Although the UNSW-NB15 and CICIDS 2017 datasets may not be the most recent, they are commonly used. Labelling network traffic datasets is challenging and error prone. The early nature of these datasets has contributed to their widespread adoption, as they are less likely to have been mislabelled or overlooked. Furthermore, they encompass a diverse range of attack types that have been thoroughly screened and processed, thereby providing intrusion detection systems with formidable evaluation tools. Owing to their long-standing importance and reliability, the UNSW-NB15 and CICIDS 2017 datasets are significant resources for intrusion detection research, even in the face of newer datasets. These datasets serve as reliable benchmarks for assessing and comparing the efficacy of various methods.

B. DATA PREPROCESSING

Fig. 1 illustrates the procedure for data preprocessing. We considered the UNSW-NB15 dataset [54] as an example. It comprises a large number of captured packets from real network traffic, covering various common types of network attacks. To enable the model to extract features, a preprocessing step was required to transform the raw CSV files into windowed sequences. Specifically, the traffic data from $(t + 1)$ to $(t + window\ size)$ are grouped into a feature packet.

1) CLASS IMBALANCE

Unbalanced datasets are common in practical application problems, particularly in fields such as medical diagnosis, information retrieval, and fraud detection. In these cases, the number of samples for each category of data can vary

TABLE 1. Features of UNSW-NB15 dataset.

No.	Features	Types	No.	Features	Types	No.	Features	Types
1	srcip	non-numeric	18	Dpkts	numeric	35	ackdat	numeric
2	sport	numeric	19	swin	numeric	36	is_sm_ips_ports	numeric
3	dstip	non-numeric	20	dwin	numeric	37	ct_state_ttl	numeric
4	dsport	numeric	21	stcpb	numeric	38	ct_flw_http_mthd	numeric
5	proto	non-numeric	22	dtcpb	numeric	39	is_ftp_login	numeric
6	state	non-numeric	23	smeansz	numeric	40	ct_ftp_cmd	numeric
7	dur	numeric	24	dmeansz	numeric	41	ct_srv_src	numeric
8	sbytes	numeric	25	trans_depth	numeric	42	ct_srv_dst	numeric
9	dbytes	numeric	26	res_bdy_len	numeric	43	ct_dst_ltm	numeric
10	sttl	numeric	27	Sjit	numeric	44	ct_src_ltm	numeric
11	dttl	numeric	28	Djit	numeric	45	ct_src_dport_ltm	numeric
12	sloss	numeric	29	Stime	numeric	46	ct_dst_sport_ltm	numeric
13	dloss	numeric	30	Ltime	numeric	47	ct_dst_src_ltm	numeric
14	service	non-numeric	31	Sintpkt	numeric	48	attack_cat	non-numeric
15	Sload	numeric	32	Dintpkt	numeric	49	Label	numeric
16	Dload	numeric	33	tcprtt	numeric			
17	Spkts	numeric	34	synack	numeric			

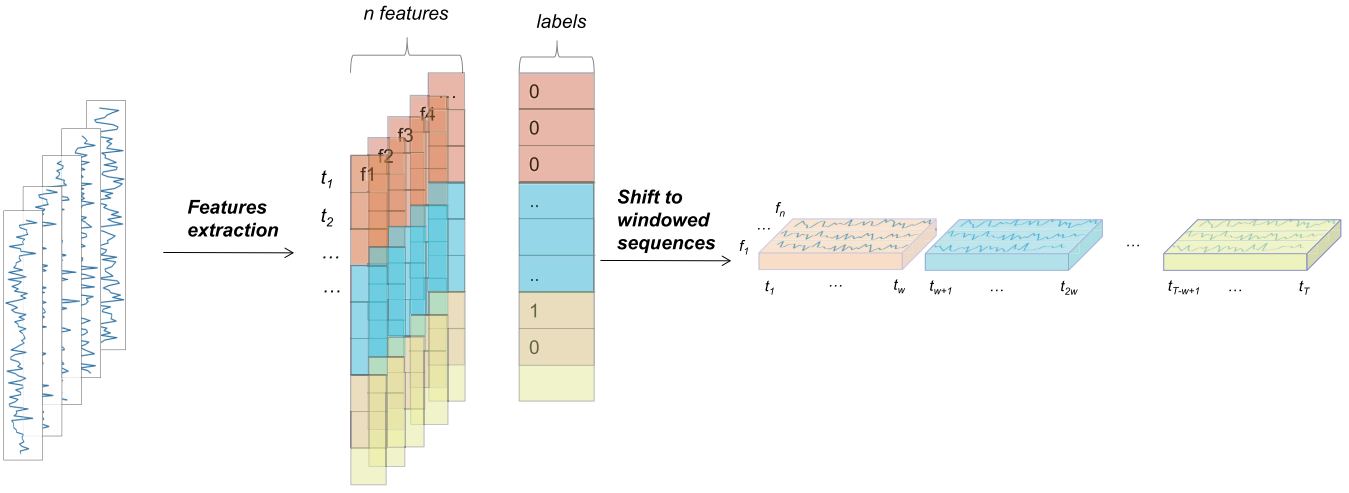


FIGURE 1. The schematic of data preprocessing. The CSV files of the original UNSW-NB15 dataset were preprocessed and transformed into window sequences using a sliding window approach to facilitate the extraction of temporal feature information for inputting into the model.

greatly, with a smaller number of negative samples than the overall sample size. The issue of imbalanced intrusion samples is particularly prevalent in intrusion traffic data sets. Benign data in the dataset often exceeded 90%, and the proportion of abnormal samples was extremely low. A high class imbalance in the dataset introduces biases that favor the majority class (benign), making the classification of minority classes challenging. There are several methods mentioned in the literature that can tackle the issue of class imbalance. Several methods have been proposed in the literature to address the issue of class imbalance. Some of these methods are as follows:

a: UNDER-SAMPLING AND OVER-SAMPLING

The method of under-sampling involves reducing the number of samples of the majority class to balance the number of samples between different categories when dealing with a large amount of data. This is typically achieved by randomly selecting and removing majority class samples [56]. In contrast, the Over-sampling method aims to prevent the loss of valuable data by generating additional samples based on the unique characteristics of a few selected samples. This approach adjusts the proportions of different samples until they reach a state of equilibrium. The two most commonly used techniques for oversampling are Synthetic Minority

Oversampling Technique (SMOTE) [57] and Adaptive Synthetic Sampling (ADASYN) [58].

b: CLASS WEIGHT STRATEGY

To give minority classes the attention they deserve during model training, weights are added to the loss function, which imposes a heavier penalty for misclassifying minority classes than for majority classes [59]. By incorporating the class weight approach, resampling of the training set is not required, making it an effective solution for addressing the issue of class imbalance in datasets.

c: SAMPLE WEIGHT STRATEGY

The sample weight approach aims to correct the class imbalance by assigning a weight to each training sample that compels each batch of data to be proportionally distributed in accordance with the desired balance during training. These weights are computed to ensure that the model considers the significance of each sample based on its weight, thereby paying more attention to the underrepresented samples. This technique effectively equalizes the influence of various classes without altering the actual distribution of the data in the training set.

To increase the overall applicability of the data and maintain its original characteristics and distribution, this paper used an oversampling technique to join the minority attack types with similar features, rather than altering or generating new composite data. This method does not require intricate algorithms or additional computing resources, resulting in a more balanced training environment.

2) DATA CLEANING

Network traffic data may contain lost packets or missing feature information as well as a significant amount of noise, outliers, and missing values. Consequently, removing these elements from data can enhance their quality and availability. In our processing, we eliminated features with minimal variance and extremely high similarity, as well as duplicate features from the dataset. Referencing the CICIDS2017 dataset, we eliminated 21 features after processing them. Table 2 outlines the specific features that were removed.

3) DATA STANDARDIZATION

To guarantee the versatility of the model in handling datasets of varying sizes and conditions,it is essential to perform necessary data preprocessing. Firstly, it is necessary to standardize the naming format. One such process involves replacing characters such as '/' and ' ' with '_' in the feature name. This is carried out to make the names more easily processed by programs and to standardize the naming conventions across each file. Network traffic data often consist of numerous features, each having distinct units. Normalization is an effective technique for eradicating the influence of these unit variances, thereby facilitating the comparison and analysis of the features more efficiently. The standardized formula is as follows:

$$x_{normalization} = \frac{x - Min}{Max - Min} \tag{1}$$

Algorithm 1 Data Preprocessing

Require: CSV files of dataset
Ensure: Time series data

1: Remove duplicate rows from dataset
2: Replace infinite values
3: Remove rows containing NaN
4: Merge labels of the same major attack class in dataset
5: Normalize input data
6: Convert data to One-hot Code
7: Split the network traffic data into equally sized sequences
8: Return the processed data, including data from index to index + win_size

TABLE 2. Features removed during data cleaning.

NO.	Feature	Note
1	flow_duration	Correlated
2	total_fwd/bwd_packet	Correlated
3	total_length_of_fwd/bwd_packet	Correlated
4	fwd_packet_length_max/mean	Correlated
5	bwd_packet_length_max/mean	Correlated
6	fwd_psh_flag	Correlated
7	bwd_psh_flag	No variance
8	fwd_urg_flag	No variance
9	bwd_urg_flag	No variance
10	packet_length_min/std/variance	Correlated
11	rst_flag_count	Correlated
12	cwe_flag_count	No variance
13	fwd_header_length.1	Duplicate
14	fwd_avg_bytes/bulk	No variance
15	fwd_avg_packet/bulk	No variance
16	fwd_avg_bulk rate	No variance
17	bwd_avg_bytes/bulk	No variance
18	bwd_avg_packet/bulk	No variance
19	bwd_avg_bulk rate	No variance
20	subflow_fwd/bwd_packets	Correlated
21	idle_mean/max	Correlated

4) DATA NUMERICALIZATION

As shown in Table 1, the UNSW-NB15 dataset encompasses non-numeric characteristics, whereas the neural network input necessitates numeric features. Consequently, it is crucial to transform certain non-numeric features, such as 'proto', 'state', and 'service', into numerical form. One-hot encoding was implemented in the processing. As an illustration, if the 'service' feature possesses three types of attributes, namely 'ssh', 'ftp', and 'http', and their corresponding numeric values are encoded as binary vectors (1,0,0), (0,1,0), and (0,0,1), respectively.

The data preprocessing procedure is illustrated in Algorithm 1.

C. MODEL ARCHITECTURE

1) DATA EMBEDDING

Fig. 2 illustrates the detailed process of data embedding. The model takes input $\mathbf{X}_{input} \in \mathbb{R}^{B\times T\times C}$, which is first

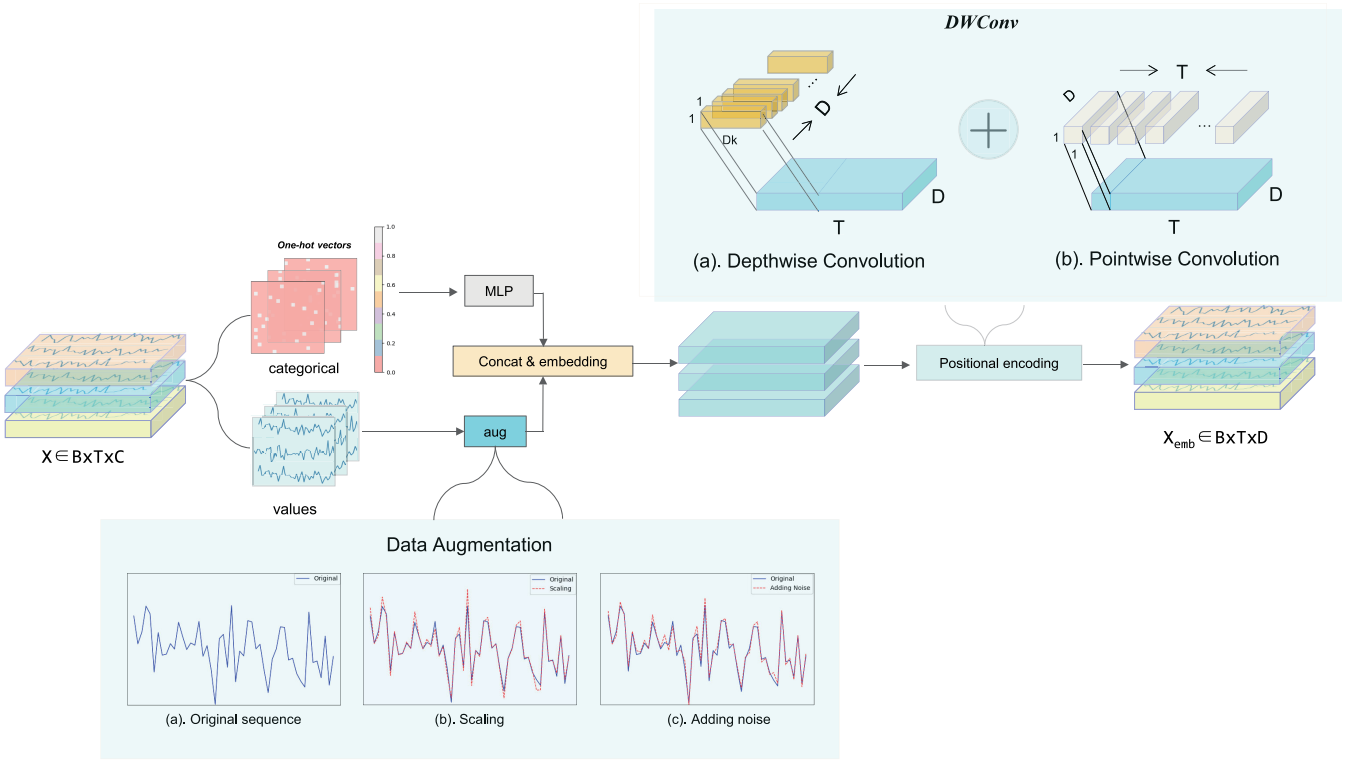


FIGURE 2. The illustration of data embedding. The network traffic data is initially partitioned into numerical and categorical data. Subsequently, each type is individually processed. The processed data is then subjected to dynamic position encoding using DWconv to obtain embedding data that facilitates effective feature extraction.

splitted into a categorical feature vector $\mathbf{X}_{cat} \in \mathbb{R}^{B\times T\times C_1}$ and a numerical feature vector $\mathbf{X}_{value} \in \mathbb{R}^{B\times T\times C_2}$, where $C = C_1 + C_2$. On the one hand, we utilize a multi-layer perceptron (MLP) to extract features for one-hot vectors $\mathbf{X}_{cat}$. On the other hand, data augmentation techniques such as jitter, scale, and permutation are applied to the numerical vectors $\mathbf{X}_{value}$. Then, the output of the MLP is concatenated with the augmented numerical features, *i.e.*,

$$\mathbf{X} = \text{Concat}(\text{MLP}(\mathbf{X}_{cat}), \text{Aug}(\mathbf{X}_{value})). \tag{2}$$

Then, $\mathbf{X}$ is multiplied by the learnable embedding matrix $\mathbf{E} \in \mathbb{R}^{B\times C\times D}$, which is used in the Embedding layer. This is followed by the patching operation, which is an essential technique to capture local features. By partitioning network traffic data into patches along the dimension C, it enables the extraction of local information. Finally, DWconv is used instead of positional embedding. DWconv consists of Depthwise Convolution and Pointwise Convolution. The Depthwise Convolution performs a separate convolution operation on each input channel, learning the correlation of each channel independently. The Pointwise Convolution then combines the results of the Depthwise Convolution along the channel dimension. Thus, the embedding of input $\mathbf{X}$ is defined as the equation 3.

$$\text{Embedding}(\mathbf{X}) = \text{DWconv}(\text{Patch}(\text{MatMul}(\mathbf{X}, \mathbf{E}))). \tag{3}$$

Depthwise convolution is employed to effectively extract critical features, mapping the extracted spatial features to lower-dimensional representations, thereby reducing computational burden and the number of network parameters. Depthwise separable convolution substantially decreases the computational complexity of convolution operations. In comparison to traditional convolution operations, it demands fewer parameters and computations, making it more feasible for processing large-scale network traffic data.

D. MODEL

The overall architecture of TSSAN is illustrated in Fig 3. As network traffic data contain less semantic information compared to text data, our proposed model aggregates information in shallow layers using convolution blocks, which contain pointwise and depthwise separable convolution layers. In deep layers, time-space self-attention blocks are used to extract information from both the temporal and spatial dimensions.

1) CONVOLUTION BLOCK

In the shallow layers, Pointwise Convolution was used to linearly combine the channels, followed by Depthwise Convolution to extract information from the data. The extracted information is then combined in the channel dimension using Pointwise Convolution. This architecture is designed to efficiently extract information from shallow

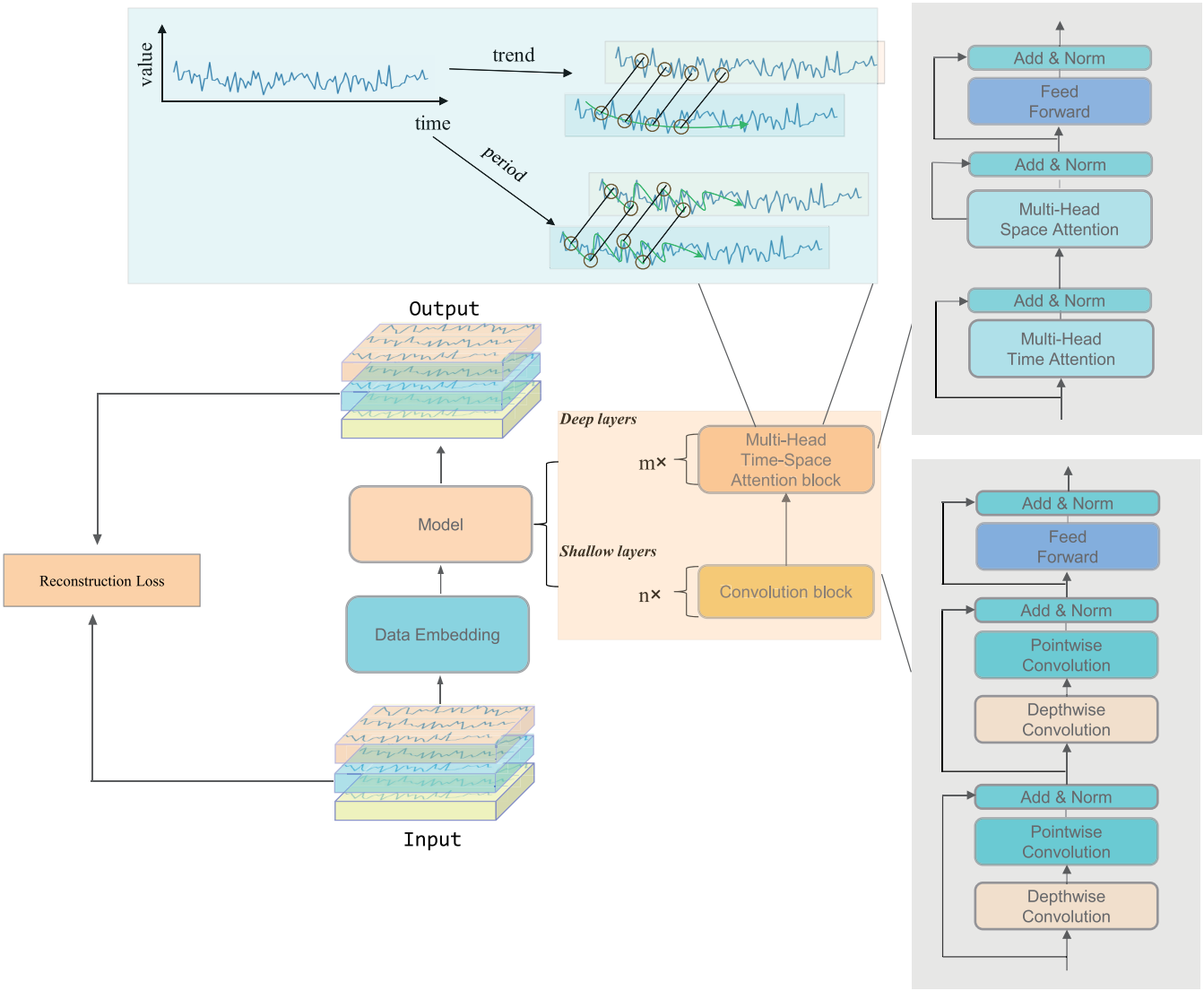


FIGURE 3. The overall architecture of the proposed model TSSAN. Depthwise convolution and pointwise convolution are employed in shallow layers to aggregate information. Space-time self-attention mechanisms are utilized in the deep layers to calculate self-attention separately in the temporal and spatial dimensions. This enables the model to effectively extract both temporal and spatial information from time-series data, such as temporal trends and periodic components and the fusion of features within and between periods.

layers and qucikly obtain effective features. These features are subsequently utilized in self-attention computations, enabling more comprehensive and accurate information processing.

The combination of Pointwise Convolution and Depthwise Convolution enables enhanced information flow between channels. The use of Depthwise Convolution reduces computational costs and parameter size by applying different filters for each channel. This approach not only accelerates computation but also reduces memory consumption. Furthermore, skip connections are used between all layers to facilitate information exchange across layers. The proposed architecture aims to achieve efficient and effective feature extraction in both the shallow and deep layers.

As illustrated in Fig 4, depthwise separable convolution plays a crucial role in the preprocessing phase of network traffic data. It excels in extracting essential features, capturing spatial information, reducing computational complexity, and enhancing model generalization. Consequently, it effectively supports the task of detecting network traffic anomalies.

2) TIME-SPACE ATTENTION BLOCK

a: THE COMPUTATION OF QUERY, KEY AND VALUE

Within each time-space attention block, a query, key, and value vector should be computed as input to the encoder. Specifically, the output of the $(\ell - 1)$-th layer is linearly transformed to serve as the input to the ℓ-th layer. This ensures that the learned representations in the previous layer are appropriately incorporated into the current layer's

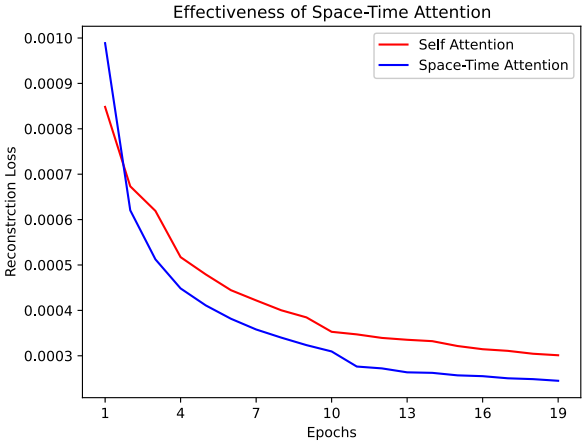


FIGURE 4. The comparison involving the use of space-time attention in terms of reconstruction loss.

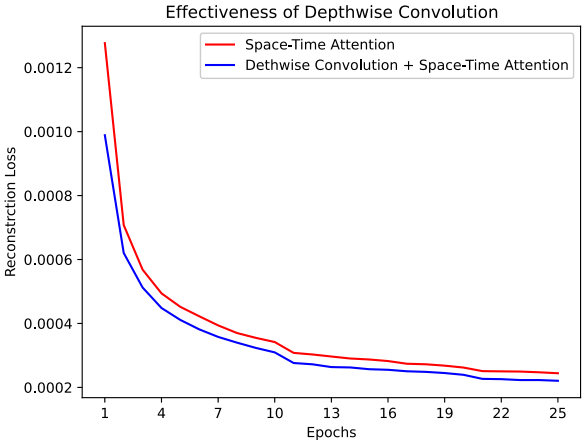


FIGURE 5. The comparison involving the use of depthwise separable convolution in terms of reconstruction loss.

computations. The query, key, and value vectors are then used to compute self-attention across both the temporal and spatial dimensions. The query, key, and value vectors are defined as the equation 4, 5 and 6.

$$Q^{(\ell)} = W_Q^{(\ell)}\mathrm{LN}\left(z^{(\ell-1)}\right) \in \mathbb{R}^{d_q}, \tag{4}$$

$$K^{(\ell)} = W_K^{(\ell)}\mathrm{LN}\left(z^{(\ell-1)}\right) \in \mathbb{R}^{d_k}, \tag{5}$$

$$V^{(\ell)} = W_V^{(\ell)}\mathrm{LN}\left(z^{(\ell-1)}\right) \in \mathbb{R}^{d_v}, \tag{6}$$

where $z^{(\ell-1)}$ represents the output of the $(\ell-1)$-th layer, and LN denotes the LayerNorm operation, which is a normalization technique that rescales the values of the input tensor to have zero mean and unit variance across each channel dimension. It has been widely used to improve the stability and convergence of the training process. LayerNorm is employed to enhance the performance of the Time-Space Attention Block. The weight matrix $W_Q^{(\ell)}$, $W_K^{(\ell)}$ and $W_V^{(\ell)}$ is used to map the input into the query, key, and value vectors.

b: TIME-SPACE SELF-ATTENTION COMPUTATION

To compute temporal self-attention, the original input vector $X_{(B,T,P)} \in \mathbb{R}^D$ is resized to $X_{(B\times P,T)} \in \mathbb{R}^D$, i.e., $X_{(B\times P)} \in \mathbb{R}^{T\times D}$, and then linearly transformed to $Q_t, K_t, V_{t(B\times P)} \in \mathbb{R}^{T\times D}$. In this context, B represents the batch size, T denotes the length of the time series, P refers to the number of patches, and D indicates the dimensions. $X_{(B\times P)}$ represents the concatenation of X along the dimensions B and P. The temporal self-attention weights α are then computed by the equation 7.

$$\alpha = \mathrm{softmax}\left(\frac{Q_t K_t^\top}{\sqrt{D}}\right) \in \mathbb{R}^{T\times T}. \tag{7}$$

Then, the temporal self-attention can be represented by the equation 8.

$$Attention(Q_t, K_t, V_t) = \alpha^{T\times T} V^{T\times D} \in \mathbb{R}^{T\times D}. \tag{8}$$

Similarly, by resizing $X_{(B,T,P)} \in \mathbb{R}^D$ as $X_{(B\times T,P)} \in \mathbb{R}^D$, $Q_s, K_s, V_s (B\times T) \in \mathbb{R}^{P\times D}$ are obtained, and spatial self-attention is computed as the equation 9.

$$Attention(Q_s, K_s, V_s) = \mathrm{softmax}\left(\frac{Q_s K_s^\top}{\sqrt{D}}\right) V \in \mathbb{R}^{P\times D}. \tag{9}$$

Finally, the temporal and spatial self-attention values were weighted and fused together to compute the time-space self-attention value.

Separating the computation of temporal and spatial features allows for an effective capture of both temporal and spatial characteristics. For instance, it captures temporal trends and periodic components within sequences, as well as the inter-feature correlations across different spatial dimensions. This approach efficiently combines the variations in feature characteristics within each period and across various periods, resulting in a significant enhancement in feature extraction efficacy.

As shown in Fig 5, the addition of temporal self-attention has led to a significant reduction in the model reconstruction loss. This indicates that the introduction of spatio-temporal self-attention mechanism enables a more comprehensive and accurate handling of network traffic data, especially when considering spatiotemporal relationships. The strength of this architecture lies in its ability to enhance feature extraction, reduce false positive rates, adapt to various attack types, and improve the model's generalization capability. This makes spatiotemporal self-attention a powerful tool for addressing network intrusion detection tasks.

IV. EXPERIMENTAL ANALYSIS AND DISCUSSION

The experiments were performed using a server equipped with the CentOS Linux release 7.8.2003 (Core) operating system, 16GB of RAM, an Intel(R) Xeon(R) Gold 5220R CPU @2.20GHz and an NVIDIA RTX 3090 graphics card. The models were implemented in Python version 3.8.12, utilizing the PyTorch version 1.11.0 library. The detailed

TABLE 3. Experimental environment.

Environment	Value
Operating system	CentOS Linux release 7.8.2003 (Core)
Processor	Intel(R) Xeon(R) Gold 5220R CPU @2.20GHz
GPU	NVIDIA RTX 3090
RAM	16GB
Programing language	Python 3.8.12
Deep learning framework	PyTorch 1.11.0

TABLE 4. The training parameters of the model.

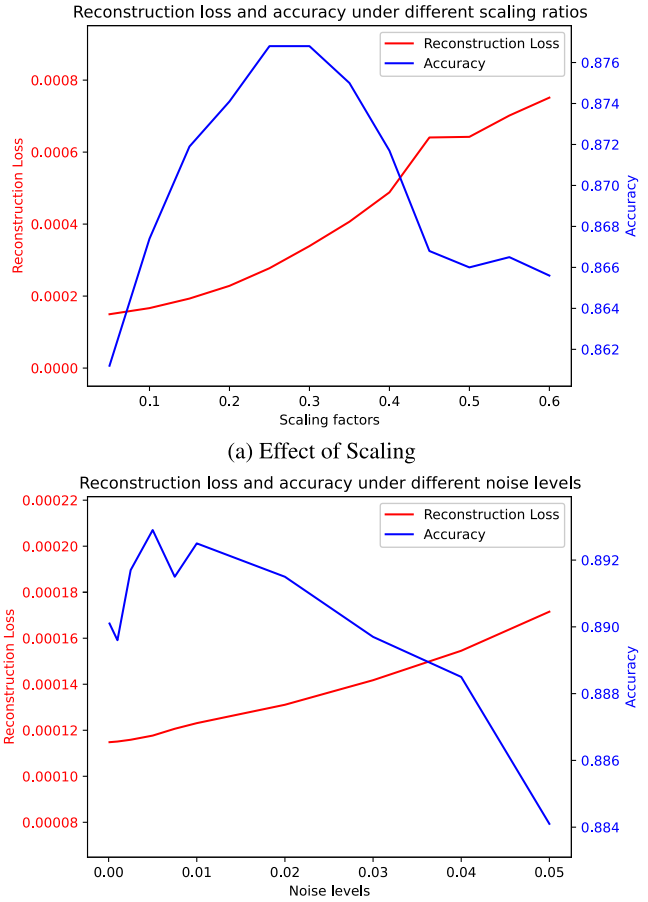
Parameter	Unsupervised learning	Fine-tuning
Epochs	20	50
Learning rate	0.0001	0.0001
Batch size	16	32
Optimizer	Adam	Adam
Loss function	MSE Loss	Cross-Entropy Loss

experimental parameter configuration is presented in Table 3 and the training parameters of the model are shown in Table 4

A. UNSUPERVISED LEARNING

In unsupervised learning anomaly detection, the reconstruction loss is commonly used to determine anomalies. Following the training of the model, each sample from the same dataset was inputted into the trained model, and the original data were reconstructed from the model's output. The reconstruction loss, which is the difference between the original data and reconstructed data, was computed for each sample. This is typically measured using various loss functions such as the mean squared error (MSE) or cross-entropy loss. A higher reconstruction loss indicates larger errors in sample reconstruction, which may suggest that it is an anomaly. To identify anomalous samples, the distribution of reconstruction losses in the training data was analyzed to determine an appropriate threshold. Generally, samples with reconstruction losses above the threshold are considered anomalies.

Table 5 presents a comparison between TSSAN and unsupervised intrusion detection methods. In this section, we evaluate the performance of the model using the ROC-AUC metric, which is a widely accepted method for assessing the effectiveness of binary classification models, particularly in the fields of machine learning and statistics. The ROC-AUC value is a single numerical value that provides a comprehensive assessment of the model's ability to distinguish between positive and negative classes. A higher ROC-AUC value, closer to 1, signifies superior discrimination abilities, as the model exhibits a higher true-positive rate and a lower false-positive rate across various threshold values. Conversely, an ROC-AUC value closer to 0.5 indicates a weak discriminative ability, akin to random guidance. The experimental outcomes indicate that TSSAN outperforms conventional machine learning techniques in unsupervised anomaly detection. This is attributable to the integration of self-attention mechanisms in both time and space, which enables the model to capture



(a) Effect of Scaling

(b) Effecting of adding noise

FIGURE 6. Effects of Data Augmentation: Although the reconstruction loss of the model continues to increase, there has been an improvement in accuracy. This indicates that the data augmentation technique has enhanced the model's generalization ability and reduced the risk of overfitting.

the general features of network traffic during unsupervised learning, thereby achieving favorable results in intrusion detection.

Table 6 presents a comparison between TSSAN and widely used unsupervised anomaly detection methods for time-series data. The evaluation metrics include precision, recall, and F1-score. Precision refers to the proportion of samples classified as positive that are actually positive. Recall refers to the proportion of samples correctly classified as positive examples among all the actual positive examples. F1 Score is the harmonic mean of Precision and Recall, and its formula is:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

The results consistently showed that TSSAN achieved a higher performance in terms of anomaly detection than the existing time-series feature extraction models. According to the results, unsupervised learning enables the model to identify intrinsic patterns and structures in the data without the need for labelled target variables. This suggests that the data possesses inherent characteristics that allow the

TABLE 5. ROC-AUC comparison with unsupervised intrusion detection methods. The experimental results of the comparative method for intrusion detection are showed on [60].

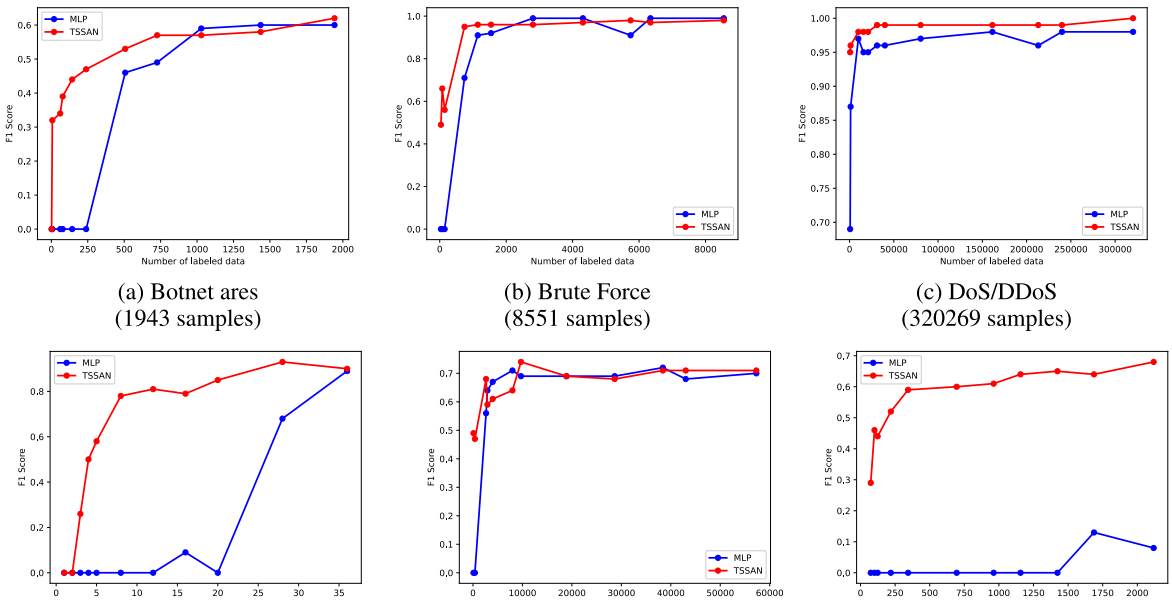
Datasets	Elliptic	LOF	I-Forest	DAlert	RCF	MSTREAM	TSSAN
UNSW	0.25	0.49	0.84	0.80	0.45	0.86	**0.90**
CICIDS2017	0.75	0.50	0.73	0.61	0.83	0.93	**0.94**

TABLE 6. Performance comparison with time-series anomaly detection mehtods.

	UNSW-NB15			CICIDS2017		
Methods	precision	recall	f1score	precision	recall	f1score
Transformer(2017) [44]	0.80	0.73	0.77	0.88	0.81	0.84
Auroformer(2021) [61]	0.83	0.76	0.79	**0.95**	0.80	0.87
Informer(2021) [49]	0.85	0.79	0.82	0.88	0.80	0.84
Reformer(2020) [48]	0.82	0.75	0.78	0.86	0.78	0.82
TimesNet(2023) [50]	0.81	0.77	0.79	0.84	0.71	0.77
TSSAN	**0.88**	**0.84**	**0.86**	0.88	**0.97**	**0.92**

TABLE 7. Composition of the CICIDS2017 Dataset. The training dataset contains a total of 2 million network flows, with approximately 80% of the flows representing normal traffic and the remaining 20% representing different types of attacks.

Category	Size		Train		Test	
Benign	2035505	83.91%	678318	82.89%	1357187	84.43%
Botnet ARES	1943	0.08%	726	0.09%	1217	0.08%
Brute Force	8551	0.35%	2806	0.34%	5745	0.36%
DoS/DDoS	320269	13.20%	107127	13.09%	213142	13.26%
Infiltration	36	0.00%	20	0.00%	16	0.00%
PortScan	57305	11.82%	28650	3.50%	28655	1.78%
Web Attack	2118	0.09%	695	0.08%	1423	0.09%

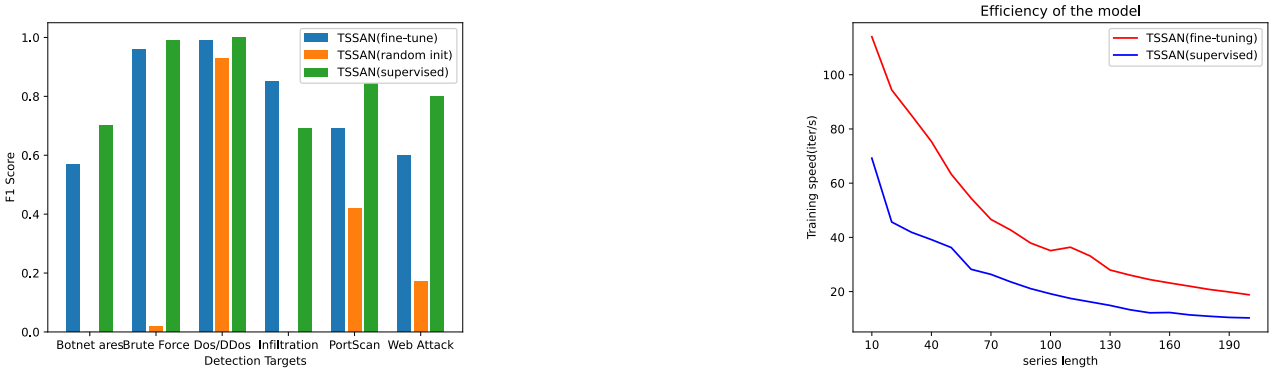


(a) Botnet ares (1943 samples) (b) Brute Force (8551 samples) (c) DoS/DDoS (320269 samples)

(d) Infiltration (36 samples) (e) PortScan (57305 samples.) (f) Web Attack (2118 samples)

FIGURE 7. The performance of TSSAN (fine-tuning) and MLP is compared across different attack types under varying quantities of labelled data.

model to make accurate predictions or classifications without the aid of external labels. Network traffic data exhibit inherent features that enable models to accurately predict or classify data without relying on external labels. By employing



(a) F1 score

(b) Runtime vs F1 score

FIGURE 8. (a) The detection performance comparison of the TSSAN(fine-tuning), TSSAN(random initialized), and TSSAN(supervised) across different attack types. (b) The comparison of model detection speed with increasing time series length.

the extraction and fusion of temporal and spatial self-attention, the model successfully uncovered potential features of network traffic, thereby facilitating in-depth understanding and analysis of the data without clear labels.

Fig 6 illustrates the impact of the Gaussian jitter and noise. Given the immense volume of network traffic, noise contamination inevitably affects the data. Therefore, applying data augmentation to numerical data can effectively mitigate the impact of noise in real-world applications, thereby significantly enhancing detection performance. As depicted in the figure, the addition of Gaussian jitter and Gaussian noise significantly enhanced the generalization ability of the model and reduced the risk of overfitting.

B. FINE-TUNING

Table 7 presents the distribution of attack samples for different categories in the CICIDS2017 dataset. During the fine-tuning process, the training and testing sets were divided in a ratio of 1:2. This strategy aims to assess the model's performance on different categories of attack samples, thereby gaining deeper insights into their capability to detect various types of anomalies. This evaluation is crucial for evaluating the robustness and feasibility of the model in real-world applications.

To assess the performance of TSSAN, we conducted a comparison with the recent methodology using the same dataset. Table 8 provides a comprehensive summary of the accuracy outcomes of various strategies on the CICDIDS2017 and UNSW-NB15 datasets. As shown in Table 8, TSSAN outperformed most existing methods on the CICDIDS2017 and UNSW-NB15 datasets. On the CICDIDS2017 dataset, our method achieved an accuracy of 99.95%, which is substantially higher than that of other methods. For the UNSW-NB15 dataset, our method also attained a remarkable accuracy of 99.20%. The experimental results indicate that the TSSAN can effectively extract spatiotemporal features, retain crucial features, and enhance the accuracy and dependability of the model.

Table 9, Table 10 and Table 11 present the detection performance of MLP, DBN, and TSSAN, respectively, for

TABLE 8. Comparison with recent methods using the same dataset.

Year	Algorithm	Dataset	Accuracy
2020	CNN-LSTM [62]	CICIDS2017	98.07%
2021	SAE-SVM [63]	CICIDS2017	99.48%
2021	SAE+Attention-BiLSTM [64]	UNSW-NB15	99.41%
2021	CAE-OCSVM [65]	UNSW-NB15	94.28%
2022	RTIDS [66]	CICIDS2017	99.35%
2022	MANDA [67]	CICIDS2017	98.41%
2022	Recurrent DL [68]	UNSW-NB15 CICIDS2017	99% 99%
2023	AE-LSTM [69]	UNSW-NB15	93.21%
This paper	TSSAN	UNSW-NB15 CICIDS2017	**99.20%** **99.95%**

different attacks. The experimental results show that these three methods perform well on frequent attack types in the dataset, such as DDoS and brute-force attacks. However, when it comes to rare attack types in the training set, such as infiltration and web attacks, TSSAN exhibits significant advantages over MLP and DBN. This indicates that the TSSAN has significantly enhanced its ability to extract general features from network traffic after unsupervised pre-training, resulting in excellent detection performance for rare anomalies.

The experimental results further validated the superiority of the TSSAN in intrusion detection tasks. Through unsupervised learning of network traffic data, TSSAN can capture the underlying correlations and feature differences among different attack types, thereby demonstrating a higher sensitivity and accuracy in detecting rare anomalies. In contrast, traditional methods, such as MLP and DBN, are limited by their feature extraction capabilities and model assumptions, leading to limited performance when confronted with rare anomalies.

Fig 7 illustrates the detection performance of the TSSAN for different proportions of the labelled training data. The experimental results demonstrate that the TSSAN exhibits a significant improvement in detection performance when

TABLE 9. Detection performance of MLP. It exhibits limitations in accurately detecting rare attack types, such as Infiltration and Web Attacks, which have minimal presence in the training set. In such cases, these rare attack instances are often misclassified as normal instances.

	Predicted							
Actual	Benign	Botnet ARES	Brute Force	DoS/ DDoS	Infiltration	Port Scan	Web Attack	**Recall**
Benign	1345312	47	16	6709	0	5078	0	99.13%
Botnet ARES	794	423	0	0	0	0	0	34.76%
Brute Force	86	0	5649	0	0	10	0	98.33%
DoS/DDoS	7504	0	0	205636	0	0	0	96.48%
Infiltration	16	0	0	0	0	0	0	0.00%
PortScan	15601	27	0	55	0	22714	0	59.16%
Web Attack	1416	0	0	7	0	0	0	0.00%
Precision	98.15%	85.11%	99.72%	96.81%	0.00%	81.70%	0.00%	

TABLE 10. Detection performence of DBN [70]. The detection performance is significantly better for attack types that are prevalent in the training set, while it deteriorates considerably for rare attack types.

	Predicted							
Actual	Benign	Botnet ARES	Brute Force	DoS/ DDoS	Infiltration	Port Scan	Web Attack	**Recall**
Benign	1355623	0	0	957	0	607	0	99.88%
Botnet ARES	1217	0	0	0	0	0	0	0.00%
Brute Force	5699	0	0	46	0	0	0	0.00%
DoS/DDoS	1217	0	0	211925	0	0	0	99.43%
Infiltration	16	0	0	0	0	0	0	0.00%
PortScan	42	3	0	93	0	38262	0	99.64%
Web Attack	1408	0	0	15	0	0	0	0.00%
Precision	99.30%	0.00%	0.00%	99.48%	0.00%	98.44%	0.00%	

TABLE 11. Detection performance of the proposed TSSAN. It demonstrates strong detection capabilities for both common and rare attack types.

	Predicted							
Actual	Benign	Botnet ARES	Brute Force	DoS/ DDoS	Infiltration	Port Scan	Web Attack	**Recall**
Benign	1350438	327	81	1182	2	4939	193	99.50%
Botnet ARES	600	617	0	0	0	0	0	50.70%
Brute Force	291	0	5452	0	0	2	0	94.90%
DoS/DDoS	2358	0	0	210764	0	12	6	98.89%
Infiltration	2	0	0	0	14	0	0	87.50%
PortScan	15467	3	32	48	1	22838	0	59.49%
Web Attack	722	0	0	5	0	2	694	48.77%
Precision	98.58%	65.15%	97.97%	99.42%	82.35%	82.17%	77.72%	

the labelled training data are limited. The experimental outcomes demonstrate that when confronted with limited labelled data, the model can effectively extract features from both temporal and spatial dimensions through the exploitation of potential structures and patterns in the data following extensive unsupervised learning. This enables the model to acquire effective feature representations of unlabelled data. These learned feature representations have the ability to capture the intrinsic patterns of the data more accurately and exhibit superior performance on limited labelled data compared to conventional supervised learning methods such as MLP. In this scenario, unsupervised learning allows the model to perform effective feature learning in the absence of labelled data, thereby enhancing its generalization capacity and overall performance. This underscores the significance of unsupervised learning in dealing with data scarcity or incomplete labelling and highlights its potential to minimize reliance on substantial volumes of labelled data.

C. RUNTIME

Fig 8 illustrates the comparison of the training and detection efficiency between supervised learning and fine-tuning after pre-training as the length of the time series increases. TSSAN(random initialiazed) refers to randomly initializing the parameters of the model's backbone network and then freezing them, with classification being performed solely by the final layer of the Multi-Layer Perceptron (MLP). The experimental results indicate that TSSAN (fine-tuned), which only trains the final classification layer in the frozen backbone network, exhibits nearly identical detection performance to the fully supervised learning model TSSAN (supervised). Nevertheless, in terms of training time, the fine-tuned model demonstrated significantly higher

efficiency than the supervised model. It is evident that pre-training on large-scale data has a significant impact on the model's detection efficiency. Fine-tuning only the classification layer has effectively reduced both training and detection times, providing strong support for real-time detection.

D. DISCUSSION

The proposed TSSAN model exhibits considerable advantages for unsupervised anomaly detection. By incorporating temporal and spatial self-attention mechanisms, the model captures intricate features of network traffic more comprehensively, leading to increased detection accuracy and robustness. Furthermore, the integration of deep separation convolution not only reduces the computational complexity of the model but also significantly enhances the performance of reconstruction loss, validating its crucial role in optimizing the model structure and performance. Moreover, the application of data augmentation techniques, such as Gaussian jitter and Gaussian noise, further bolsters the generalization capacity of the model, mitigates the risk of overfitting, and enables the model to and enables the model to better cope with various complex situations in the real world. In addition, the TSSAN model demonstrates strong performance in small-sample scenarios by extracting temporal and spatial features from data through unsupervised learning. These enhancements not only improve the detection performance and efficiency of the model but also broaden the potential of unsupervised learning in practical network security settings.

Despite its impressive capabilities, the model may still have certain limitations. Firstly, the pre-training phase necessitates substantial computational resources because of the requirement of unlabelled data on a large scale. Additionally, while the model has shown outstanding performance in the dataset, there are still challenges in applying it to complex network environments in real-world scenarios. The outcomes of the two data augmentation procedures demonstrate that the model possesses the capacity to handle noise and exhibits a certain extent of generalization. Consequently, the model holds great promise for identifying anomalies in real-world network settings. Therefore, additional research is required in future studies to evaluate the practicality of the model and to address the needs of practical application.

V. CONCLUSION

In this paper, an unsupervised learning model called the Time-Space Separable Attention Network(TSSAN) is proposed for network intrusion detection, aiming to improve the detection performance for intrusion detection when lacking labelled data. The experimental results demonstrate that TSSAN achieves a detection performance comparable to traditional MLP methods with less than 10% of the labelled data. This highlights the efficiency and superiority of the TSSAN in utilizing limited labelled data. Furthermore, the fine-tuning process of the TSSAN is rapid and efficient, providing strong support for real-time intrusion detection applications.

Although TSSAN exhibits remarkable performance in feature extraction from unlabelled network traffic data and the detection of a limited number of attack types, there are still some potential issues that require improvement. Future research should focus the practical applications of the TSSAN. Although TSSAN performs well on network traffic datasets, challenges may arise in complex and dynamic real-world network environments. Therefore, utilizing the TSSAN in real-world network intrusion detection systems to evaluate its performance in actual scenarios will further demonstrate the feasibility and practicality of the model.

ACKNOWLEDGMENT

The authors would like to express their gratitude to all those who contributed to this research. They extend their heartfelt thanks to their supervisor Xizhao Luo, for his guidance, support, and invaluable insights. They also thank the members of their research group for their constructive feedback and discussions.

REFERENCES

[1] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[2] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowl.-Based Syst.*, vol. 136, pp. 130–139, Nov. 2017, doi: 10.1016/j.knosys.2017.09.014.

[3] W. Meng, W. Li, and L. Kwok, "Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 3883–3895, Dec. 2015, doi: 10.1002/sec.1307.

[4] S. Afroz, S. M. A. Islam, S. N. Rafa, and M. Islam, "A two layer machine learning system for intrusion detection based on random forest and support vector machine," in *Proc. IEEE Int. Women Eng. (WIE) Conf. Electr. Comput. Eng. (WIECON-ECE)*, Dec. 2020, pp. 300–303.

[5] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A novel two-stage deep learning model for network intrusion detection: LSTM-AE," *IEEE Access*, vol. 11, pp. 37131–37148, 2023, doi: 10.1109/ACCESS.2023.3266979.

[6] X. Yang, J. Yan, W. Liao, X. Yang, J. Tang, and T. He, "SCRDet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2384–2399, Feb. 2023.

[7] K. Chowdhary and K. Chowdhary, "Natural language processing," in *Fundamentals of Artificial Intelligence*, 2020, pp. 603–649.

[8] R. A. Bridges, T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, and Q. Chen, "A survey of intrusion detection systems leveraging host data," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–35, Nov. 2020.

[9] A. Singla, E. Bertino, and D. Verma, "Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 127–140, doi: 10.1145/3320269.3384718.

[10] J. Sinha and M. Manollas, "Efficient deep CNN-BiLSTM model for network intrusion detection," in *Proc. 3rd Int. Conf. Artif. Intell. Pattern Recognit.*, Jun. 2020, pp. 223–231.

[11] R. Wang, J. Yan, and X. Yang, "Unsupervised learning of graph matching with mixture of modes via discrepancy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 10500–10518, Aug. 2023.

[12] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., Dec. 2020, pp. 1–4. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North*, Minneapolis, MN, USA, 2019, pp. 4171–4186, doi: 10.18653/v1/n19-1423.

[14] B. Xu, L. Sun, X. Mao, C. Liu, and Z. Ding, "Strengthening network security: Deep learning models for intrusion detection with optimized feature subset and effective imbalance handling," *Comput., Mater. Continua*, vol. 78, no. 2, pp. 1995–2022, 2024. [Online]. Available: http://www.techscience.com/cmc/v78n2/55548

[15] H. Nizam, S. Zafar, Z. Lv, F. Wang, and X. Hu, "Real-time deep anomaly detection framework for multivariate time-series data in industrial IoT," *IEEE Sensors J.*, vol. 22, no. 23, pp. 22836–22849, Dec. 2022.

[16] H. Khazane, M. Ridouani, F. Salahdine, and N. Kaabouch, "A holistic review of machine learning adversarial attacks in IoT networks," *Future Internet*, vol. 16, no. 1, p. 32, Jan. 2024. [Online]. Available: https://www.mdpi.com/1999-5903/16/1/32

[17] X. Yang and J. Yan, "On the arbitrary-oriented object detection: Classification based approaches revisited," *Int. J. Comput. Vis.*, vol. 130, no. 5, pp. 1340–1365, May 2022.

[18] R. Wang, J. Yan, and X. Yang, "Neural graph matching network: Learning Lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5261–5279, Sep. 2022.

[19] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, Sep. 2023.

[20] P. Jairu and A. B. Mailewa, "Network anomaly uncovering on CICIDS-2017 dataset: A supervised artificial intelligence approach," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (eIT)*, Mankato, MN, USA, May 2022, pp. 606–615, doi: 10.1109/eIT53891.2022.9814045.

[21] C. Lu, Y. Cao, and Z. Wang, "Research on intrusion detection based on an enhanced random forest algorithm," *Appl. Sci.*, vol. 14, no. 2, p. 714, Jan. 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/2/714

[22] P. Singh, J. J. P, A. Pankaj, and R. Mitra, "Edge-detect: Edge-centric network intrusion detection using deep neural network," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2021, pp. 1–6, doi: 10.1109/CCNC49032.2021.9369469.

[23] R. Kumar, A. Aljuhani, D. Javeed, P. Kumar, S. Islam, and A. K. M. N. Islam, "Digital twins-enabled zero touch network: A smart contract and explainable AI integrated cybersecurity framework," *Future Gener. Comput. Syst.*, vol. 156, pp. 191–205, Jul. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X24000608

[24] D. Javeed, T. Gao, M. S. Saeed, and M. T. Khan, "FOG-empowered augmented intelligence-based proactive defensive mechanism for IoT-enabled smart industries," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18599–18608, Aug. 2023.

[25] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2017, pp. 1024–1034. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

[26] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, May 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[27] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del-Rincón, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 876–889, Jun. 2020, doi: 10.1109/TNSM.2020.2971776.

[28] L. Yang and A. Shami, "IoT data analytics in dynamic environments: From an automated machine learning perspective," *Eng. Appl. Artif. Intell.*, vol. 116, Nov. 2022, Art. no. 105366, doi: 10.1016/j.engappai.2022.105366.

[29] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*, 2019, pp. 703–716, doi: 10.1007/978-3-030-30490-4_56.

[30] M. A. Bashar and R. Nayak, "TAnoGAN: Time series anomaly detection with generative adversarial networks," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Canberra, ACT, Australia, Dec. 2020, pp. 1778–1785, doi: 10.1109/SSCI47803.2020.9308512.

[31] J. Cui, L. Zong, J. Xie, and M. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," *Appl. Intell.*, vol. 53, no. 1, pp. 272–288, Jan. 2023, doi: 10.1007/s10489-022-03361-2.

[32] Z. Zhong, C. Xie, and X. Tang, "Intrusion traffic detection and classification based on unsupervised learning," *IEEE Access*, vol. 12, pp. 67860–67879, 2024, doi: 10.1109/ACCESS.2024.3400213.

[33] P. F. de Araujo-Filho, M. Naili, G. Kaddoum, E. T. Fapi, and Z. Zhu, "Unsupervised GAN-based intrusion detection system using temporal convolutional networks and self-attention," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 4, pp. 4951–4963, Aug. 2023, doi: 10.1109/TNSM.2023.3260039.

[34] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.

[35] M. Catillo, A. Pecchia, and U. Villano, "CPS-GUARD: Intrusion detection for cyber-physical systems and IoT devices using outlier-aware deep autoencoders," *Comput. Secur.*, vol. 129, Jun. 2023, Art. no. 103210, doi: 10.1016/j.cose.2023.103210.

[36] T. Zhang, W. Chen, Y. Liu, and L. Wu, "An intrusion detection method based on stacked sparse autoencoder and improved Gaussian mixture model," *Comput. Secur.*, vol. 128, May 2023, Art. no. 103144, doi: 10.1016/j.cose.2023.103144.

[37] I. O. Lopes, D. Zou, I. H. Abdulqadder, F. A. Ruambo, B. Yuan, and H. Jin, "Effective network intrusion detection via representation learning: A denoising AutoEncoder approach," *Comput. Commun.*, vol. 194, pp. 55–65, Oct. 2022, doi: 10.1016/j.comcom.2022.07.027.

[38] G. Li and J. J. Jung, "Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges," *Inf. Fusion*, vol. 91, pp. 93–102, Mar. 2023, doi: 10.1016/j.inffus.2022.10.008.

[39] W. Alahamade, I. Lake, C. E. Reeves, and B. De La Iglesia, "A multi-variate time series clustering approach based on intermediate fusion: A case study in air pollution data imputation," *Neurocomputing*, vol. 490, pp. 229–245, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221017963

[40] C.-Y. Hsu and W.-C. Liu, "Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing," *J. Intell. Manuf.*, vol. 32, no. 3, pp. 823–836, Mar. 2021, doi: 10.1007/s10845-020-01591-0.

[41] M. Dutt, S. Redhu, M. Goodwin, and C. W. Omlin, "SleepXAI: An explainable deep learning approach for multi-class sleep stage identification," *Appl. Intell.*, vol. 53, no. 13, pp. 16830–16843, Jul. 2023, doi: 10.1007/s10489-022-04357-8.

[42] Y. Choi, H. Lim, H. Choi, and I.-J. Kim, "GAN-based anomaly detection and localization of multivariate time series data for power plant," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 71–74.

[43] M. Adiban, S. M. Siniscalchi, and G. Salvi, "A step-by-step training method for multi generator GANs with application to anomaly detection and cybersecurity," *Neurocomputing*, vol. 537, pp. 296–308, Jun. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223003065

[44] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, May 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[46] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4091–4098. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16325

[47] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *Proc. 10th Int. Conf. Learn. Represent. (ICLR)*, Apr. 2022. [Online]. Available: https://openreview.net/forum?id=LzQQ89U1qm_

[48] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020. [Online]. Available: https://openreview.net/forum?id=rkgNKkHtvB

[49] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, May 2021, pp. 11106–11115. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17325

[50] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," 2022, *arXiv:2210.02186*.

[51] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Montreal, QC, Canada, Aug. 2021, pp. 2352–2359, doi: 10.24963/ijcai.2021/324.

[52] Z. Yue et al., "Ts2vec: Towards universal representation of time series," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 8, pp. 8980–8987. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20881

[53] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 813–824. [Online]. Available: http://proceedings.mlr.press/v139/bertasius21a.html

[54] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[55] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.

[56] B. Liu and G. Tsoumakas, "Dealing with class imbalance in classifier chains via random undersampling," *Knowl.-Based Syst.*, vol. 192, Mar. 2020, Art. no. 105292, doi: 10.1016/j.knosys.2019.105292.

[57] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[58] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Hong Kong, Jun. 2008, pp. 1322–1328, doi: 10.1109/IJCNN.2008.4633969.

[59] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 4368–4374, doi: 10.1109/IJCNN.2016.7727770.

[60] S. Bhatia, A. Jain, P. Li, R. Kumar, and B. Hooi, "MStream: Fast anomaly detection in multi-aspect streams," in *Proc. Web Conf.*, Apr. 2021, pp. 3371–3382, doi: 10.1145/3442381.3450023.

[61] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. NIPS*, vol. 34, Dec. 2021, pp. 22419–22430. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html

[62] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[63] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 387–403, Jun. 2021, doi: 10.1007/s10207-020-00508-5.

[64] G. Lu and X. Tian, "An efficient communication intrusion detection scheme in AMI combining feature dimensionality reduction and improved LSTM," *Secur. Commun. Netw.*, vol. 2021, pp. 1–21, Apr. 2021, doi: 10.1155/2021/6631075.

[65] A. Binbusayyis and T. Vaiyapuri, "Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM," *Appl. Intell.*, vol. 51, no. 10, pp. 7094–7108, Oct. 2021, doi: 10.1007/s10489-021-02205-9.

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

[67] N. Wang, Y. Chen, Y. Xiao, Y. Hu, W. Lou, and Y. T. Hou, "MANDA: On adversarial example detection for network intrusion detection system," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 2, pp. 1139–1153, Mar. 2023, doi: 10.1109/TDSC.2022.3148990.

[68] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108156, doi: 10.1016/j.compeleceng.2022.108156.

[69] H. C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Eng. Sci. Technol., Int. J.*, vol. 38, Feb. 2023, Art. no. 101322. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098622002312

[70] O. Belarbi, A. Khan, and P. Carnelli, "An intrusion detection system based on deep belief networks," in *Proc. Int. Conf. Sci. Cyber Secur.* Switzerland: Springer, 2022, pp. 377–392, doi: 10.1007/978-3-031-17551-0_25.

RUI XU is currently pursuing the degree in computer science and technology with the School of Computer Science and Technology, Soochow University. His research interests primarily include the fields of artificial intelligence and network security, with a focus on intrusion detection.

QI ZHANG was born in 1998. He is currently pursuing the master's degree. He is a member of China Computer Federation. His main research interests include machine learning and artificial intelligence security.

YUNJIE ZHANG received the B.S. degree from Suzhou University of Science and Technology. He is currently pursuing the M.S. degree with Soochow University. His research interest includes to construct and apply Lie group convolutional neural networks to image processing.