IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 24, 2022, accepted June 7, 2022, date of publication June 13, 2022, date of current version June 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3182333

RTIDS: A Robust Transformer-Based Approach for Intrusion Detection System

ZIHAN WU[1], HONG ZHANG[1], PENGHAI WANG[1], AND ZHIBO SUN[2]
[1]School of Cyber Security and Computer, Hebei University, Baoding 071000, China
[2]Department of Computer Science, Arizona State University, Tempe, AZ 85281, USA

Corresponding author: Hong Zhang (hzhang@hbu.edu.cn)

This work was supported in part by the Science and Technology Research Project of Hebei Higher Education Institutions under Grant QN2020133, and in part by the Natural Science Foundation of Hebei Province of China under Grant F2019201361.

ABSTRACT Due to the rapid growth in network traffic and increasing security threats, Intrusion Detection Systems (IDS) have become increasingly critical in the field of cyber security for providing secure communications against cyber adversaries. However, there exist many challenges for designing a robust, efficient and accurate IDS, especially when dealing with high-dimensional anomaly data with unforeseen and unpredictable attacks. In this paper, we propose a Robust Transformer-based Intrusion Detection System (RTIDS) reconstructing feature representations to make a trade-off between dimensionality reduction and feature retention in imbalanced datasets. The proposed method utilizes positional embedding technique to associate sequential information between features, then a variant stacked encoder-decoder neural network is used to learn low-dimensional feature representations from high-dimensional raw data. Furthermore, we apply self-attention mechanism to facilitate network traffic type classifications. Extensive experiments reveal the effectiveness of the proposed RTIDS on two publicly available real traffic intrusion detection datasets named CICIDS2017 and CIC-DDoS2019 with F1-Score of 99.17% and 98.48% respectively. A comparative study with classical machine learning algorithm support vector machine (SVM) and deep learning algorithms that include recurrent neural network (RNN), fuzzy neural network (FNN), and long short-term memory network (LSTM) is conducted to demonstrate the validity of the proposed method.

INDEX TERMS Intrusion detection, feature representation, self-attention mechanism, transformer.

I. INTRODUCTION

Nearly 61% of the global population are active Internet users as of July 2021 [1]. Despite the fact that the Internet offers enormous conveniences and opportunities to people, it is a platform for criminals to launch illicit attacks through the networks, resulting in a loss of $600 billion in 2017 [2]. To mitigate the threats, researchers have proposed various Network Intrusion Detection Systems (NIDS) [3]–[6]. Li *et al.* [7] presented an approach to classify attack categories and normal traffic by using the support vector machine (SVM), and Thaseen *et al.* [8] utilized the Random Tree to detect malicious network activities. However, given the problems of the large volume of data with complicated feature representations, these approaches cannot detect network attacks effectively [9]. Deep learning has begun to gain more attention from the cybersecurity community because it has been widely used to process large-scale datasets in natural language processing (NLP) and image processing [10]. Loukas *et al.* [11] proposed a cyber-physical intrusion detection system based on recurrent neural network architecture and deep multilayer perceptron, and Otoum *et al.* [12] developed a clustered intrusion detection system in wireless sensor networks based on the restricted Boltzmann machine. Although many deep learning based NIDSs have achieved desirable detection performance [13], there are still three significant challenges that remain unresolved:

- It is difficult to obtain prior knowledge through past hidden states while training detection models [14].
- It is difficult to retain as many pivotal traffic features as possible when compressing intrusion detection datasets [15]
- Deep learning models such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) are expensive to train, and their neural networks are too deep [16], [17].

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

In recent years, self-attention mechanism based models such as transformer and its variants have obtained huge success in text classification, dialogue recognition, machine translation, and other natural language processing (NLP) tasks. The essence of transformer is to pre-train on a vast text corpus first then applied the trained model on a smaller task-dedicated dataset to fine tune [18]. Transformer has the advantage of computation efficiency and scalability, it has also been used in the field of image classification and computer vision [19].

Enlightened by transformer's ability to handle ordered sequences of data, Huang *et al.* [20] used a variant of transformer to detect system log anomalies and proved its robustness on unstable log data. Bikmukhamedov *et al.* [21] applied a transformer model to classify traffic data and obtained good result.

In this paper, the main goal of our research is to propose a robust transformer-based intrusion detection system (RTIDS) that is able to process a large volume of complicated raw network data efficiently and provide effective detection performance. RTIDS consists of three modules and features an innovative hierarchy self-attention design inspired by the Transformer model [22]. Specifically, we apply input and positional embedding to convert input network traffic into fixed-dimension vectors as input representations. Then we use stacked encoders and decoders for feature extraction and learning the contextual relations between inputs. Since the input features have different impacts on the classification result, we utilize self-attention mechanism to learn different weights of those feature representations. Additionally, the performance of RTIDS was compared to that of four other mainstream machine learning and deep learning algorithms using two popular intrusion detection evaluation datasets (CICIDS2017 and CIC-DDoS2019), and RTIDS has shown significant performance improvement in comparison with existing intrusion detection methods.

The main contributions of this article can be summarized as follows:

- We present an innovative Robust Transformer-based Intrusion Detection System (RTIDS), which efficiently extracts and transforms high-dimensional raw data into low-dimensional representations.
- The RTIDS can effectively balance the dimensionality reduction and feature retention in highly imbalanced and high-dimensional datasets.
- We design a self-attention mechanism that can capture contextual information between network traffic features for detecting intrusions.
- The RTIDS has a much higher detection performance than other popular intrusion detection models.

The rest of this article is organized as follows: section 1 is the introduction of our study. Section 2 summarizes related works. In section 3, we illustrate the background information and the proposed model in detail. Section 4 presents the experimental setup used in our study. In section 5, we evaluate the experimental results and make a comparative analysis with another four intrusion detection methods. Finally, section 6 concludes this article.

II. RELATED WORK

To detect and prevent networks from cyber attacks, researchers have invested much effort in proposing various Network Intrusion Detection Systems (NIDS) [23], [24]. In general, NIDS has two main categories: signature-based IDS and anomaly-based IDS.

Signature-based IDS detects threats by comparing the network activities with the known indicator of compromise (IOC), which could be a hash value of a file, a malicious IP address, etc. [25]. Chowdhury *et al.* [26] suggested combining two machine learning algorithms together to classify signature-based intrusions. Their work applied a simulated annealing technique to generate three random feature sets and an SVM algorithm to identify anomalous behaviors. Veeraiah *et al.* [27] built a trust-aware signature-based IDS to identify potential intrusions in the MANET nodes using trust tables. In their study, they utilized fuzzy Naive Bayes and fuzzy clustering algorithms. Both He [28] and Sutskever [29] designed signature-based routing protocols for detecting Sybil attacks that exist in the Internet of Things. However, due to the nature of detecting known threats, the outdated and limited sources of IOCs, nowadays, a considerable number of sophisticated cyber attacks can bypass the signature-based IDS easily.

In contrast, anomaly-based intrusion detection systems compare all network activities with a pre-trained and normalized baseline that presents how the system normally behaves, which makes the system be able to detect unknown malicious network activities. Alamiedy *et al.* [30] proposed an enhanced anomaly-based IDS model based on multi-objective grey wolf optimization (GWO) algorithm. The GWO algorithm was employed as a feature selection mechanism to identify the most relevant features from the NSL-KDD dataset, which contributed to high classification accuracy. Furthermore, a support vector machine was used to estimate the capability of selected features in predicting attacks accurately. Satam *et al.* [31] presented a Wireless Intrusion Detection System (WIDS) to detect attacks on Wi-Fi networks. In this approach, they used n-grams to model the normal behavior of the Wi-Fi protocol and used Random Forest, AdaBoost, and other machine learning methods to classify Wi-Fi traffic flows. Gothawal *et al.* [32] formulated a game-theoretic model-based anomaly Intrusion Detection System (IDS) to detect the RPL attacks and verify their malicious activities. The proposed approach consists of two interrelated formulations, such as a stochastic game for attack detection and an evolutionary game for attack confirmation.

To further empower the IDS to automatically detect network intrusions, Saeed *et al.* [33] applied artificial neural network (ANN), probabilistic neural network, and chi-square algorithms to detect distributed denial of service (DDoS) attacks. Mahmood *et al.* [34] combined a log tracking model and a spatio-temporal ML model to build

an anomaly detection framework. However, due to the high-dimensional presentations of the raw data nowadays, these machine learning-based models cannot efficiently process the data in a timely manner and have a poor detection performance [35], [36].

In recent years, deep learning techniques has been widely used in the field of intrusion detection and has achieved remarkable results. Moustafa *et al.* [37] proposed an adversarial statistical learning mechanism that applied Outlier Dirichlet Mixture-based ADS (ODM-ADS) method to detect abnormal behaviors for the KDD-99 dataset. Jiang *et al.* [38] developed a novel dense random neural network that included a hidden Markov model (HMM) algorithm to detect network attacks. Iranmanesh, Saeid, *et al.* [39] proposed a heuristic distributed scheme (HIDE) to validate the mobility pattern of vehicles and identify malicious vehicles by penalizing or rewarding vehicles based on the contacts' conformation. Maseer *et al.* [40] applied 10 popular supervised and unsupervised machine learning algorithms for identifying effective and efficient anomaly-based IDS (AIDS) networks and computers. Their models were tested by using a recent and highly unbalanced multiclass CICIDS2017 dataset that involves real-world network attacks. Moreover, Abdulhammed *et al.* [41] implemented a web attack detection system based on distributed edge devices, which employed multiple concurrent learning models to improve stability and performance. Furthermore, the researcher also found that recurrent neural network (RNN) based IDS can effectively detect network intrusions and identify attack types on the NSL-KDD dataset [42], [43].

Although those deep learning-based models can improve detection performance dramatically [44], some challenges cannot be overlooked. RNN model is too slow to take full advantage of modern fast computing devices and tends to forget long-term information. In order to solve this problem, Chung *et al.* [45] proposed Gated Recurrent Unit (GRU) method to help examine long time series. While CNN model easily loses much valuable information due to the pooling layer and requires a large dataset. Ding *et al.* [46] presented Asymmetric Convolution Block (ACB), an architecture-neutral structure as a CNN building block, which uses 1D asymmetric convolutions to strengthen the square convolution kernels. Their model can help retain important feature information and also reduce sample dependency. Wang *et al.* [47] designed a hybrid neural network structure called DDosTC that combined transformers and a convolutional neural network (CNN) to detect distributed denial-of-service (DDoS) attacks on software-defined network (SDN) and tested on the dedicated DDoS testbed dataset CICDDoS2019.

III. BACKGROUND AND PROPOSED FRAMEWORK

In this section, we explore the background knowledge of the transformer model and the detailed structure and processing mechanisms of the proposed robust Transformer-based intrusion detection system (RTIDS). At first, We introduce the related background information about the transformer model, then we present the general framework for the proposed method. Lastly, we illustrate the concrete design of the detection model.

A. TRANSFORMER MODEL

The Transformer model is similar to most competitive neural sequence conduction models, and it also uses an encoder-decoder structure. The encoder maps the input symbolic representation sequence $(x^1, x^2, \ldots, x^n)$ into a continuous representation sequence $\mathrm{z} = (z^1, z^2, \ldots, z^n)$, the decoder outputs the given continuous representation sequence as $(y^1, y^2, \ldots, y^m)$. Every step of the model is autoregressive, which means the output generated by each encoder or decoder is the input of the next encoder or decoder except the input of the first encoder at the bottom of the encoder stack [48], [49]. Transformer model aims at transforming the input feature sequence to the corresponding vector representations. A major difference between transformer and RNN is a self-attention mechanism that utilizes attention matrices instead of the recurrent connection [50].

B. RTIDS FRAMEWORK

We propose an innovative hierarchical transformer structure-based IDS for efficiently processing complex network traffic data without losing critical details. Fig.1 shows the overall framework of RTIDS.

RTIDS consists of three components: data preparation module, RTIDS model construction module, and real-time intrusion detection module. As part of the data preparation module, raw network traffic data are processed through four steps: data cleaning, data normalization, feature selection, and dataset splitting. Afterward, we train a variant Transformer model that is fine-tuned to detect abnormal network activities in the RTIDS model construction module. Three units comprise the real-time intrusion detection module. The network connection unit is responsible for receiving and forwarding network packets, and the intrusion detection unit takes advantage of the well-tuned RTIDS model to detect suspicious network activities. Depending on the type of attacks detected, the mitigation unit employs predetermined strategies in order to minimize the system's risk.

C. MODEL DESIGN

The detection model is the brain of RTIDS, and it is composed of three components: input embedding, encoder and decoder stacks, and the softmax layer. Typically, the model presents all raw inputs with equal-length vectors by using input embedding. Encoders and decoders then process these vectors via mechanisms of masked multi-head self-attention in order to integrate information of the inputs and help enhance the model. Lastly, the softmax layer is used to calculate the malicious probability of the network activities. Algorithm 1 illustrates the pseudocode description of the proposed framework.

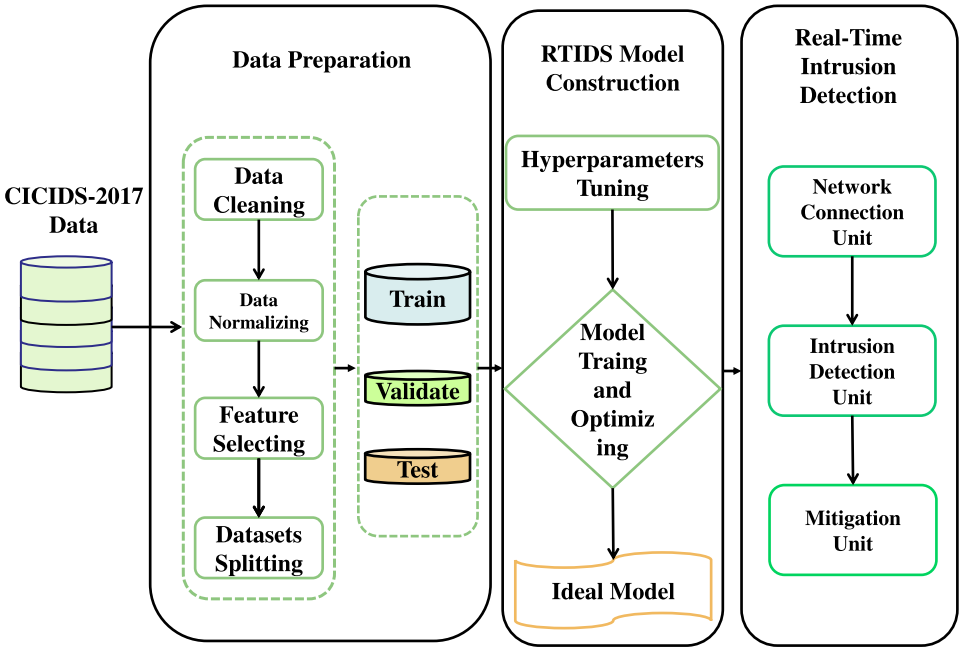

FIGURE 1. The framework of RTIDS. RTIDS consists of three modules: the data preparation module, the model construction module and the real-time intrusion detection module.

Algorithm 1 RTIDS Algorithm

Input: Training set $S = (x\_i, y\_i), i = 1, 2, \ldots N$, x_i is the network traffic sample, y_i is the corresponding label

Output: Classification probabilities of the predicted class

1: **for** $i \leftarrow 0$ *until* *numOfEpochs* **do**
2: **for** *Sample s*:*Batch* **do**
get its vectorized representation s_r
put s_r into encoder and decoder stacks for feature extraction and selection
use transformerModel.MultiHeadAttention function to calculate the attention scores of features
use transformerModel.SoftMax function to get classification probabilities
3: **end for**
use stochastic gradient descent (SGD) algorithm to minimize loss function
4: **end for**

1) INPUT EMBEDDING

In our model, we first apply embedded techniques to process the raw data inputs. Due to network traffic contains some sequential feature information such as source and destination port number and IP address quadruple. Our model includes additional positional encoding information at the bottom of the encoder stack to take advantage of the input features' location information. Since the positional embedding dimensions are identical to input embedding dimensions, the final input of the self-attention layer is the summation of embedding and positional encoding. Moreover, the input embedding uses the sine function and cosine function with different frequencies to implement positional encoding. In particular, the odd positions are encoded with a cosine function, and the even positions are encoded with a sine function, shown in Eq. (1) and Eq. (2).

$$PE(pos, 2i) = sin(pos/1000^{2i/d_{model}}) \quad (1)$$

$$PE(pos, 2i+1) = cos(pos/1000^{2i/d_{model}}) \quad (2)$$

2) ENCODER AND DECODER STACK

a: ENCODER STACK

There are six encoders in the encoder stack, and each encoder is composed of a multi-head self-attention network and a point-wise feed-forward network (FFN). The dimension of FFN layers is a hyperparameter that can be adjusted during training. For the optimal performance, we assign 1024 neurons in FFN layers and set 32 as the padding size of embedding. Additionally, residual connection and layer regularization are used to calculate the results of each sub-layer, which are then passed to the next encoder in the stack (see Fig. 2).

b: DECODER STACK

We add an additional multi-headed masked self-attention sub-layer into each decoder in contrast to the original transformer

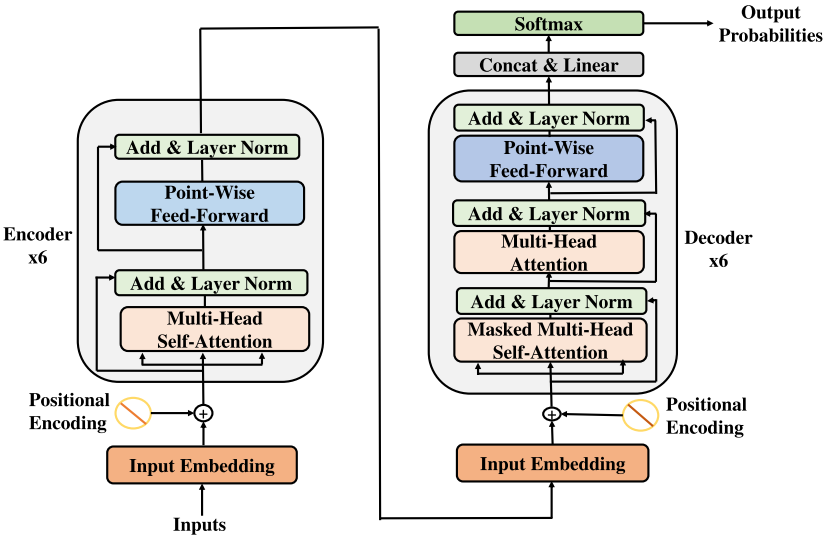

FIGURE 2. The overall structure of the improved transformer model. The input features first undergo positional embedding then are sent to multi-head self-attention sublayer. After that, the calculated result is processed by the normalization sublayer and point-wise feed-forward sublayer.

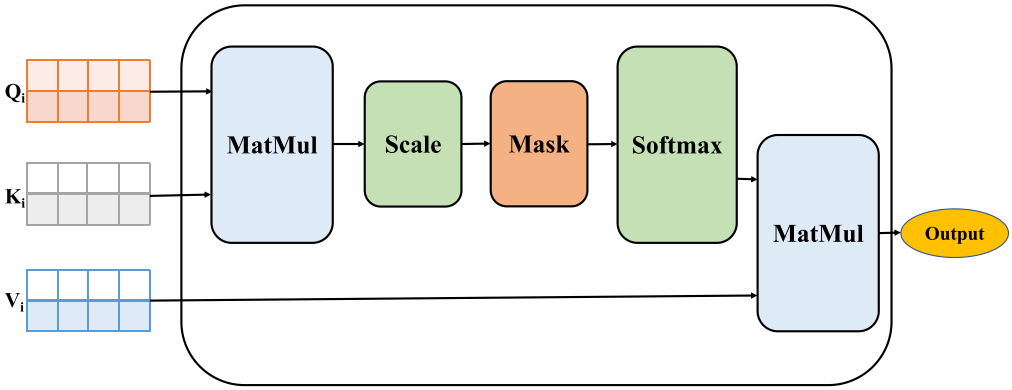

FIGURE 3. Masked scaled dot-product attention, Q_i, K_i, V_i indicate query, key and value matrix respectively.

model. In order to improve the robustness of the proposed intrusion detection model, we mask a portion of features randomly and then predict them using other unmasked features. Additionally, to maintain the hierarchical structure of our model, we deploy six decoders in the decoder stack. After reconstructing encoder and decoder stacks, we apply the softmax layer as the final output layer for classification. As depicted in Fig.2, there is a residual connection between input's self-attention sublayer and point-wise feed-forward network, followed by layer normalization. The purpose of these techniques is to improve the model's performance by addressing potential problems, such as the vanishing gradient and the covariate shift. In addition to the attention sublayer, each encoder and decoder also contains a fully connected sublayer called point-wise feed-forward network. It contains two linear transformations activated by the ReLU function σ, see Eq. (3) for illutration. Furthermore, identical weights are used for each row of the attention matrices, which can be considered as a convolution on every row of the attention-transformed matrix. It can be viewed that this step enriches the embeddings with additional information.

$$FFN(x) = \sigma(max(0, xW_1 + b_1)W_2 + b_2) \tag{3}$$

3) MASKED SELF-ATTENTION MECHANISM

The masked scaled dot-product attention mechanism for head h in decoder is shown in Fig.3. It takes input as a set of queries(Q), keys(K) and values(V). Masked scaled dot-product attention can be computed with Eq. (4), where d_k is the scaling factor, and M is the masking matrix:

$$Attention(Q, K, V) = softmax(M + \frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

In Eq. (4), the division by $\sqrt{d_k}$ stabilizes the gradients during training. After that, $M \in R^{k \times k}$ is used to prevent attending subsequent positions, which ensures that the prediction for

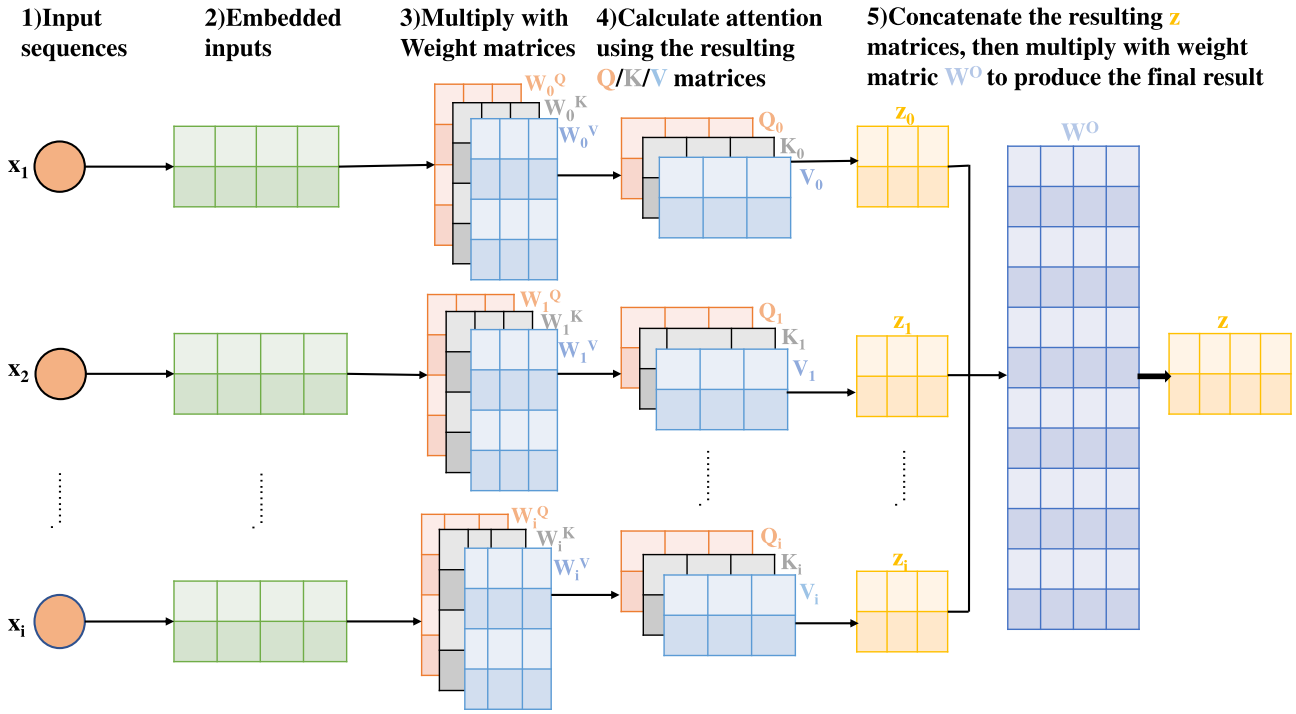

FIGURE 4. The multi-head attention mechanism, specifically, the raw input x_i is first embedded using embedding functions, then, x_i is multiplied with weight matrices W^Q, W^K, W^V respectively, and the queries, keys and values are linearly projected i times with different, learned linear projections to d_q, d_k and d_v dimensions respectively. The attention function is then performed in parallel on each of these projected versions of queries, keys and values, yielding i output values. These values are concatenated and projected again, resulting in the final values.

position i depends only on the known outputs at positions less than i. This mechanism can help tackle the overfitting problem. Moreover, the masking operation is performed in the softmax function by adding $-\infty$. After that, each row of the matrix is then normalized into a probability distribution using the softmax activation function. Finally, a new input representation is constructed via the dot product of the normalized matrix and V. The performance of the self-attention layer can be further improved by the multi-head mechanism, illustrated in Fig.4. With multi-head attention, each attention head independently maintains its own Q/K/V weight matrix. The calculation process is similar to the single head attention, shown in Eqs (5) and (6). Note that $head_i$ denotes the ith attention head of the self-attention sublayer. Moreover, the dimensions of the parameter matrix are $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$, $W_i^O \in R^{d_{model} \times hd_v}$, respectively. The results obtained by each head are concatenated together to construct the final result. This mechanism enables the model to focus on different positions and provides multiple representation sub-spaces for the attention layer.

$$\begin{aligned} &MultiHead(Q, K, V) \\ &= Concat(head_1, \ldots, head_h)W^O \qquad (5) \\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \qquad (6) \end{aligned}$$

For the purpose of stable training and faster convergence, layer normalization is applied on each sample $x \in R^d$ as Eq. (7):

$$LN(x) = \frac{x - \mu}{\delta} \cdot \alpha + \beta \qquad (7)$$

where $\mu \in R, \delta \in R$ are the mean and standard deviation of the input feature respectively, $\cdot$ is the element-wise dot operator, $\alpha \in R^d, \beta \in R^d$ are trainable affine transform parameters.

We apply Stochastic Gradient Descent (SGD) algorithm to train the model and we choose the cross-entropy as the loss function.

IV. EXPERIMENT SETUP

In this section, We first introduce the hardware and software configurations that we utilize to conduct our experiments then illustrate the dataset used in this work as well as the data preprocessing methods. At last, we explain the evaluation metrics used to show the RTIDS's performance.

A. EXPERIMENT CONFIGURATION

Our experiments are executed on a computer with 64-bit Intel Core i7 2.70 GHz processor, Nvidia GeForce RTX 2080 graphics card, 32 GB RAM, running Windows 10 Operating System. The proposed model is implemented by Pytorch 1.9.0. The version of Python we choose to execute the program is 3.7.0 and the main software packages we use include Numpy, Keras, Pandas, Sklearn and Matplotlib. We train our model with 25 training epochs, a

0.5 dropout rate, a learning rate of $5e-4$, and a batch size of 128.

B. DESCRIPTIONS OF CICIDS2017 DATASET

The intrusion detection evaluation dataset CICIDS2017 has been widely used by researchers to analyze and develop new models and algorithms since it was first introduced by the Canadian Institute for Cybersecurity (CIC) [51]. Compared with the NSL-KDD dataset, the CICIDS2017 dataset is up-to-date and offers a broader protocol and attack pool. The traffic records with 79 distinct features are divided into 15 traffic types: which are Benign, FTP-Patator, SSH-Patator, DoS Slowhttptest, DoS Hulk, DoS GoldenEye, DoS slowloris, Heartbleed, Web Attack-Brute Force, Web Attack-XSS, Web Attack-SQL Injection, Infiltration, Botnet, PortScan and DDoS [52]. The dataset spanns over eight different files containing five days normal and attacks traffic data of Canadian Institute of Cybersecurity (CIC). A brief description of those data files is shown in Table 1. We summarise the CICIDS2017 dataset in terms of attack types in Table 2. It can be seen that the dataset is highly imbalanced, resulting in a significant challenge in anomaly detection. For example, the total number of ''Benign'' traffic type in the dataset is 2,273,097 which occupies a very high proportion (80.30%), while the total number of traffic labelled ''Heartbleed'' is only 11 (<0.01%).

C. DESCRIPTIONS OF CIC-DDoS 2019 DATASET

The CIC-DDoS 2019 dataset is the latest designed dataset which was shared by the Canadian Institute for Cybersecurity (CIC), it was prepared in a proper test context and includes the result of real network traffic analyses [53]. The CIC-DDoS2019 dataset includes 30,480,823 records, including 30,423,960 DDoS attacks records and 56,863 benign records. Furthermore, the DDoS attacks are divided into 11 subtypes. Each record is described by 86 features. The statistic information and attack types of the dataset is summarized in Table 3.

The dataset is implemented with two networks, namely Attack-Network and Victim-Network. The Victim-Network is a highly security infrastructure with firewall, router, switches, and several common operating systems along with an agent that provides the benign behaviors on each PC. The Attack-Network is a completely separated third party infrastructure that executes different types of DDoS attacks.

D. DATA PREPROCESSING

Sine both CICIDS2017 and CIC-DDoS2019 datasets have high class imbablance rate (CIR), we employ Synthetic Minority Oversampling Technique (SMOTE) to increase the quantity of minority class samples by generating samples that does not exist in the original dataset. With this arrangement, it can avoid overfitting problem when constructing classfication model.

Unlike other machine learning models, our method aim to retain as many features as possible for the purpose of accuracy improvement, the self-attention mechanism in our model can select features automatically.

We transform all the symbolic features contained in the datasets into numerical values. After converting, the dataset is normalized into the range of [0-1] by using the Min-Max normalization technique which is given by Eq. (8).

$$\tilde{x} = \frac{x - min_{x_{train}}}{max_{x_{train}} - min_{x_{train}}} \tag{8}$$

Note that $max_(x_{train})$ and $min_(x_{train})$ refer to the maximum value and minimum value of a certain feature X in the training set respectively. We then split the entire dataset into the training set (70%), the validation set (15%) and the testing set (15%).

E. EVALUATION METRICS

In addition to accuracy, we also use precision, recall, and F1-score, which are widely used for anomaly detection tasks in order to evaluate model performance. These metrics can be expressed by Eqs. (9, (10), (11) and (12) respectively:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

True positive (TP) refers to abnormal network traffics that are correctly detected; false positive (FP) corresponds to normal traffics that are incorrectly classified as abnormal ones; true negative (TN) represents normal traffics correctly classified as normal, and false negative (FN) is abnormal traffics incorrectly classified as normal ones. The details of TP, FP, TN, and FN can be seen in the confusion matrix in Table 4. In the experiments, we use the training set to learn the general pattern of the traffic sequence and then predict the abnormal traffic in the test set to achieve the purpose of anomaly detection.

V. EXPERIMENT EVALUATION AND ANALYSIS

Our evaluation baselines for network traffic classification employ a classical machine learning algorithm, support vector machines (SVM), and deep learning algorithms that include recurrent neural network (RNN), fuzzy neural network (FNN), and long short-term memory (LSTM) [54], [55].

A. EVALUATION RESULTS OF CICIDS2017 AND CIC-DDoS2019 DATASETS

Table 5 shows the performance of the RTIDS model on CICIDS2017 dataset by using accuracy, precision, recall and F1-score. We find that the proposed method achieves the highest detection accuracy (99.98%) for traffic type ''Benign,'' while having a lowest performance on detecting ''SQL Injection'' (50.36%). Due to the ''SQL Injection''

TABLE 1. Description of CICIDS2017 dataset files.

Name of Files	Class Found
Monday-Hours.pcap_ISCX.csv	Benign
Tuesday-Hours.pcap_ISCX.csv	Benign, FTP-Patator, SSH-Patator
Wednesday-pcap_ISCX.csv	Benign, DoS GoldenEye, Dos Hulk, DoS lowhttptest, DoS slowloris, Heartbleed
Thursday-WebAttacks.pcap_ISCX.csv	Benign, Brute Force, SQL Injection, XSS
Thursday-Infilteration.pcap_ISCX.csv	Benign, Infiltration
Friday-pcap_ISCX.csv	Benign, Bot
Friday-PortScan.pcap_ISCX.csv	Benign, PortScan
Friday-DDoS.pcap_ISCX.csv	Benign, DDoS

TABLE 2. Summary of the CICIDS-2017 dataset.

Class Type	Flow Count	Ratio
Benign	2,273,097	80.30%
DoS Hulk	231,073	8.16%
PortScan	158,930	5.61%
DDoS	128,027	4.52%
DoS GoldenEye	10,293	0.36%
FTP-Patator	7,938	0.28%
SSH-Patator	5,897	0.21%
DoS slowloris	5,796	0.20%
DoS Slowhttptest	5,499	0.19%
Bot	1,966	0.07%
Web Attack-Brute Force	1,507	0.05%
Web Attack-XSS	652	0.02%
Infiltration	36	0.01%
Web Attack-SQL Injection	21	$< 0.01\%$
Heartbleed	11	$< 0.01\%$
Total	2,830,743	100.00%

TABLE 3. Statistic information of CIC-DDoS2019 dataset.

Class Type	Flow Count	Ratio
Benign	56,863	0.02%
DDoS_DNS	5,071,011	16.64%
DDoS_LDAP	2,179,930	7.15%
DDoS_MSSQL	4,522,492	14.84%
DDoS_NetBIOS	4,093,279	13.43%
DDoS_NTP	1,202,642	3.95%
DDoS_SNMP	5,159,870	16.93%
DDoS_SSDP	2,610,611	8.56%
DDoS_TFTP	2,082,580	6.83%
DDoS_UDP	3,134,645	10.28%
DDoS_UDP-Lag	366,461	1.20%
DDoS_WebDDoS	439	$< 0.01\%$
Total	30,480,823	100%

TABLE 4. Confusion matrix of evaluation metrics.

Type	Predicted Positive	Predicted Negative
Attack(Positive)	Predicted Attack as attack(TP)	Predicted attack as Normal(FN)
Normal(Negative)	Predicted Normal as attack(FP)	Predicted normal as Normal(TN)

sample is exceptionally scarce in the entire dataset, leading to undesirable performance for our model. The behavior pattern of traffic type "Bot" is analogous to normal network traffic, which increases the difficulty for our model to correctly identify "Bot" attacks, resulting in poor performance.

The performance of our proposed RTIDS model on CIC-DDoS2019 dataset is shown in Table 6. It can be seen that RTIDS model achieves highest detection rate on traffic type

TABLE 5. Performance of RTIDS on CICIDS2017 dataset.

Attack Type	Accuracy	Precision	Recall	F1-Score
Benign	99.98%	99.99%	99.98%	99.97%
DDoS	99.90%	99.89%	99.94%	99.91%
Infiltration	99.37%	95.21%	96.74%	95.97%
PortScan	99.92%	99.93%	99.93%	99.91%
Bot Attack	92.70%	92.98%	92.79%	92.11%
Patator-FTP	99.98%	99.50%	99.75%	99.62%
Patator-SSH	97.53%	98.70%	98.11%	98.40%
Brute Force	99.26%	99.81%	99.53%	99.67%
XSS	98.62%	98.95%	98.74%	98.84%
SQL Injection	50.36%	49.82%	49.47%	47.17%
DDoS-GoldenEye	97.87%	97.16%	97.51%	97.33%
DDoS-Hulk	99.63%	99.76%	99.69%	99.72%
DDoS-Slowhttptest	94.96%	97.36%	96.14%	96.75%
DDoS-slowloris	98.50%	98.97%	98.22%	98.59%
Heartbleed	81.96%	81.83%	81.89%	81.86%

TABLE 6. Performance of RTIDS on CIC-DDoS2019 dataset.

Attack Type	Accuracy	Precision	Recall	F1-Score
Benign	99.47%	98.79%	99.74%	99.60%
WebDDos	89.77%	88.89%	84.46%	86.95%
UDP-Lag	96.27%	95.91%	96.08%	96.09%
NTP	99.65%	99.51%	99.58%	99.58%
LDAP	98.03%	97.62%	97.32%	97.82%
SSDP	91.41%	92.00%	85.11%	90.93%
UDP	97.29%	75.71%	86.05%	85.16%
NetBIOS	94.03%	99.60%	96.79%	96.73%
MSSQL	96.69%	90.23%	93.42%	93.35%
SNMP	98.05%	93.82%	95.92%	95.89%
TFTP	97.71%	99.65%	97.51%	98.67%
DNS	97.36%	97.00%	97.04%	97.18%

"DDoS_NTP" with accuracy of 99.65%, and the detection rate for class "DDoS_WebDDoS" can also reach 89.77%. Thus, it proves that the proposed method in this research effectively improves the issue of low detection rate due to data scarceness.

B. COMPARISON ANALYSIS WITH BASELINE MODELS ON CICIDS2017 DATASET

In this section, we compare the results of our model with those of the classic machine learning model SVM and other deep learning models. The overall classification results of all models is summarized in Table 7. In terms of classification accuracy, the proposed method is this paper is 0.89%, 1.33%, 2.14% and 1.0% higher than SVM-IDS, RNN-IDS, LSTM-IDS and FNN-IDS respectively. From the

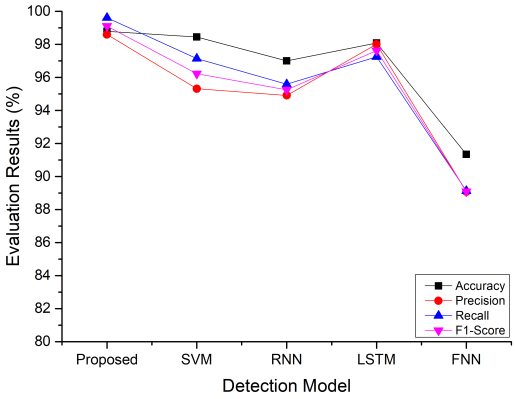

FIGURE 5. Performance of various intrusion detection models for detecting Benign traffic.

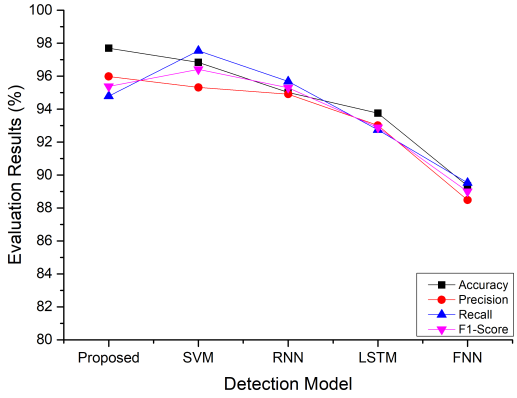

FIGURE 6. Performance of various intrusion detection models for detecting Bot attack.

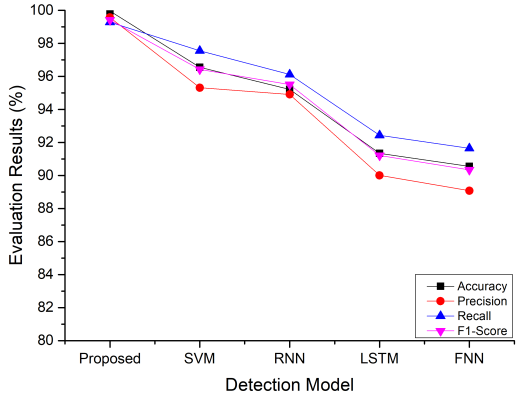

FIGURE 7. Performance of various intrusion detection models for detecting Patator attacks.

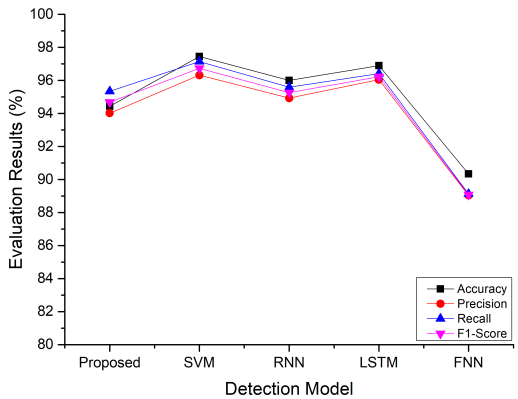

FIGURE 8. Performance of various intrusion detection models for detecting Web attacks.

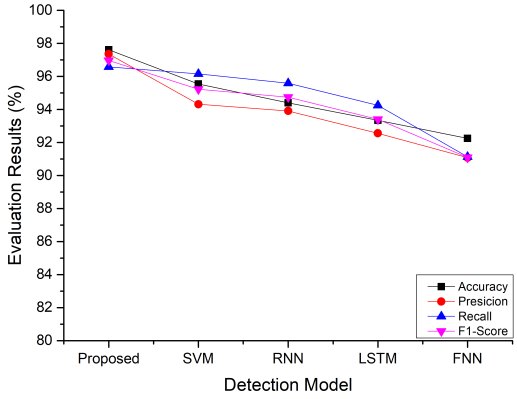

FIGURE 9. Performance of various intrusion detection models for detecting DoS attacks.

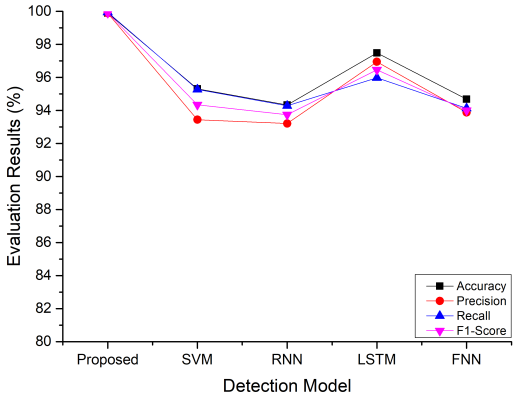

FIGURE 10. Performance of various intrusion detection models for detecting DDoS attack.

classification results, we can see that our proposed method is effective. Compared with other models, the training time of our proposed model is 195.6s. Although the training time of this model is not the shortest, but the time performance is acceptable for practical implementation.

The initial evaluation focuses on the detection performance of traffic with the label "Benign." Fig.5 shows that our model outperforms all other detection algorithms with an accuracy of 99.65%. The intrusion detection accuracy of the proposed model on traffic labeled "Bot" is 97.70%, which is a significant improvement over other models (see Fig. 6). Since the malicious traffic labeled with "FTP-Patator" and "SSH-Patator" are similar in their attack behavior and characteristics, we combine and label them "Patator Attacks,"

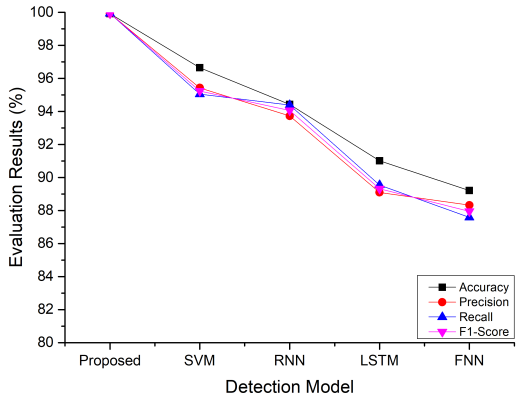

FIGURE 11. Performance of various intrusion detection models for detecting PortScan attack.

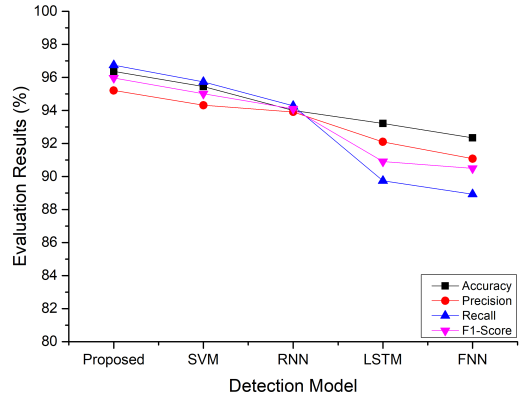

FIGURE 12. Performance of various intrusion detection models for detecting Infiltration attack.

resulting in 13,835 records with this new label. The detection performance of this type of traffic is shown in Fig.7. Compared with other models, our designed model with an accuracy of 97.56% is better at detecting "Patator Attacks." The traffic labeled "Web Attack" includes three sub-types: "Web Attack-Brute Force," "Web Attack-XSS" and "Web Attack-SQL Injection." Fig.8 shows the detection performance of different intrusion detection models for this type of traffic. While the accuracy of our proposed model is higher than other classification models, its performance metrics such as precision and recall are slightly below those of these baseline models. The reason is that it is not easy to choose a proper threshold value that provides high detection accuracy rate or false alarm rate for this traffic type in our model. We merge the traffic data labeled "DoS GoldenEye," "DoS Hulk," "DoS Slowhttptest," "DoS slowloris" and "Heartbleed" into a new dataset labeled "DoS," resulting in a total of 252,627 records. Fig.9 shows the overall detection performance of this type of data. As shown, the proposed model substantially improves the accuracy, precision, recall, and F1-score of identifying "Dos" traffic data. Due to the differences in traffic characteristics and behavior patterns between

TABLE 7. Overall classification results of all models on CICIDS2017 dataset.

Algorithms	Accuracy	Precision	Recall	F1-Score	Time (S)
SVM-IDS	98.45%	98.32%	98.73%	98.02%	193.2
RNN-IDS	98.01%	98.91%	98.29%	98.10%	227.8
LSTM-IDS	97.21%	97.10%	97.29%	97.19%	265.1
FNN-IDS	98.35%	98.98%	98.83%	98.50%	198.9
Proposed	99.35%	98.98%	98.83%	99.17%	195.6

TABLE 8. Overall classification results of all models on CIC-DDoS2019 dataset.

Algorithms	Accuracy	Precision	Recall	F1-Score	Time (S)
SVM-IDS	94.02%	94.54%	94.24%	94.88%	264.6
RNN-IDS	96.91%	96.94%	97.01%	96.45%	299.4
LSTM-IDS	97.76%	97.43%	97.71%	97.58%	306.1
FNN-IDS	95.55%	95.78%	95.63%	95.50%	298.2
Proposed	98.58%	98.82%	98.66%	98.48%	275.6

DDoS attacks and DoS attacks, for example, DoS attacks usually occur between single machines, whereas DDoS attacks typically involve a large number of computers and multiple networks. Thus, we exclusively evaluate traffic data labelled "DDoS" in the dataset. Fig.10 depicts the detection results, and we find that our model still outperforms others with an accuracy of 99.90%. Finally, we evaluate the classification performance of different models using traffic data labelled as "PortScan" and "Infiltration." Fig.11 and Fig.12 show the performance results. It can be observed that our proposed model performs better than other models on both detection tasks.

C. COMPARISON ANALYSIS WITH BASELINE MODELS ON CIC-DDoS2019 DATASET

In order to further validate the effectiveness of the proposed method in this paper, we also perform experiments on CIC-DDoS2019 dataset. The total experiment results of all classifiers are summarized in Table 8. As the table shows, the classification accuracy of our proposed method has improved by 4.56%, 1.67%, 0.81% and 3.03% compared with SVM-IDS, RNN-IDS, LSTM-IDS and FNN-IDS. However, the training time of the proposed model is still not the lowest, but since the model needs to be trained only once and can be used for off-line intrusion detection in the network, the time performance receivable.

Fig.13 gives the multi-class classification accuracy of various intrusion detection models on CIC-DDoS2019 dataset. Both classical machine learning and deep learning classifiers have achieved less ideal performance on class type "WebDDoS" and "SSDP" in comparison with other class types. During experiments, we find that the features have similar characteristics between the "WebDDoS" and "SSDP." This means that the classifiers require additional features to classify the traffic record "WebDDoS" and "SSDP" correctly. In spite of this fact, the detection accuracy of "WebDDoS" and "SSDP" of the proposed method reaches 89.77% and 91.41% respectively. Since our model incorporates as many

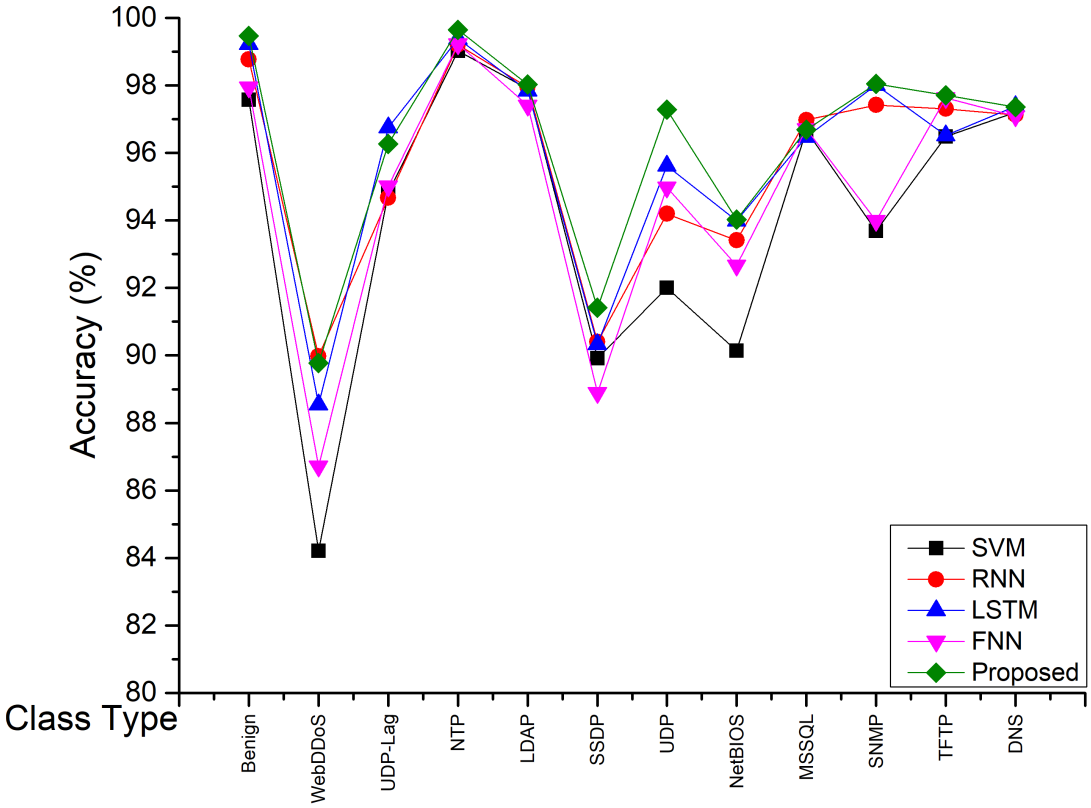

FIGURE 13. Multi-class classification accuracy of all models on CIC-DDoS2019 dataset.

features as possible during training, thus, the experiments testifies the advantage of the proposed framework.

VI. CONCLUSION AND FUTURE WORK

In this article, we propose a robust Transformer-based intrusion detection system, called RTIDS, for detecting abnormal activities and traffic violations in networks. RTIDS provides an all-in-one intrusion detection solution composed of three modules: a data preparation module, a RTIDS model-construction module, and a real-time intrusion detection module. The framework employs transformer model for feature extraction and selection. In addition, to prevent our neural network detection model from overfitting, we design a variant transformer with one additional masked multi-head self-attention sublayer in the decoder stack. We also employ the Synthetic Minority Over-sampling Technique (SMOTE) for oversampling minority class samples to combat the class imbalance issue. Furthermore, we evaluate the performance of RTIDS on the CICIDS2017 and CIC-DDoS2019 datasets. The accuracy of the RTIDS algorithm we proposed is 98.45%, and the precision is 98.32%, the Recall is 98.73%, the F1-score is 98.02% for CICIDS2017 dataset and the accuracy is 98.58%, the precision is 98.82%, the Recall is 98.66%, the F1-score is 98.45% for CIC-DDoS2019 dataset respectively. Experimental results demonstrate that RTIDS performs better than the mainstream classical and deep learning intrusion detection algorithms used in other IDSs. In terms of training time,the time performance of our proposed method is acceptable as an off-line intrusion detection tool in the network.

Our future work will focus on how to increase the speed of the transformer algorithm for the quick-response intrusion detection system in order to significantly reduce the damage caused by anomalous events. Furthermore, in the next step of our work, we will consider employ meta-learning method to tackle the few shot classification problem.

REFERENCES

[1] S. Caton and R. Landman, "Internet safety, online radicalisation and young people with learning disabilities," *Brit. J. Learn. Disabilities*, vol. 50, no. 1, pp. 88–97, Mar. 2022.

[2] J. Lewis, "Economic impact of cybercrime, no slowing down," Center Strategic Int. Stud., McAfee, San Jose, CA, USA, 2018, vol. 13, p. 2019.

[3] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12499–12514, 2020.

[4] H. Wang, Z. Cao, and B. Hong, "A network intrusion detection system based on convolutional neural network," *J. Intell. Fuzzy Syst.*, vol. 38, no. 6, pp. 7623–7637, Jun. 2020.

[5] S. Al and M. Dener, "STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102435.

[6] Z. Wu, J. Wang, L. Hu, Z. Zhang, and H. Wu, "A network intrusion detection method based on semantic re-encoding and deep learning," *J. Netw. Comput. Appl.*, vol. 164, Aug. 2020, Art. no. 102688.

[7] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424–430, Jan. 2012.
[8] S. Thaseen and C. A. Kumar, "An analysis of supervised tree based classifiers for intrusion detection system," in *Proc. Int. Conf. Pattern Recognit., Informat. Mobile Eng.*, Feb. 2013, pp. 294–299.
[9] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107784.
[10] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Proc. Comput. Sci.*, vol. 89, pp. 117–123, Jan. 2016.
[11] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, vol. 6, pp. 3491–3508, 2018.
[12] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Netw. Lett.*, vol. 1, no. 2, pp. 68–71, Jun. 2019.
[13] R. Vinayakumar, M. Alazab, K. 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.
[14] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 56–70, May 2020.
[15] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, Oct. 2019.
[16] L. Ruiz, F. Gama, and A. Ribeiro, "Gated graph recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 6303–6318, 2020.
[17] M. Xia, W. Liu, K. Wang, W. Song, C. Chen, and Y. Li, "Non-intrusive load disaggregation based on composite deep long short-term memory network," *Expert Syst. Appl.*, vol. 160, Dec. 2020, Art. no. 113669.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[19] 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," 2020, *arXiv:2010.11929*.
[20] S. Huang, Y. Liu, C. Fung, R. He, Y. Zhao, H. Yang, and Z. Luan, "HitAnomaly: Hierarchical transformers for anomaly detection in system log," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2064–2076, Dec. 2020.
[21] R. F. Bikmukhamedov and A. F. Nadeev, "Generative transformer framework for network traffic generation and classification," *T-Comm*, vol. 14, no. 11, pp. 64–71, 2020.
[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[23] S. Chormunge and S. Jena, "Efficient feature subset selection algorithm for high dimensional data," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 6, no. 4, p. 1880, Aug. 2016.
[24] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Comput. Netw.*, vol. 174, Jun. 2020, Art. no. 107247.
[25] F. Erlacher and F. Dressler, "On high-speed flow-based intrusion detection using snort-compatible signatures," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 495–506, Jan. 2022.
[26] M. N. Chowdhury, K. Ferens, and M. Ferens, "Network intrusion detection using machine learning," in *Proc. Int. Conf. Secur. Manage. (SAM), Steering Committee World Congr. Comput. Sci., Comput. Eng. Appl. Comput. (WorldComp)*, 2016, p. 30.
[27] N. Veeraiah and B. T. Krishna, "Trust-aware FuzzyClus-fuzzy NB: Intrusion detection scheme based on fuzzy clustering and Bayesian rule," *Wireless Netw.*, vol. 25, pp. 4021–4035, Jan. 2019.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3104–3112.
[30] T. A. Alamiedy, M. Anbar, Z. N. M. Alqattan, and Q. M. Alzubi, "Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 9, pp. 3735–3756, Sep. 2020.
[31] P. Satam and S. Hariri, "WIDS: An anomaly based intrusion detection system for Wi-Fi (IEEE 802.11) protocol," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 1077–1091, Mar. 2021.
[32] D. B. Gothawal and S. V. Nagaraj, "Anomaly-based intrusion detection system in RPL by applying stochastic and evolutionary game models over IoT environment," *Wireless Pers. Commun.*, vol. 110, no. 3, pp. 1323–1344, Feb. 2020.
[33] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385–393, Jan. 2016.
[34] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3854–3861.
[35] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012.
[36] H. Luo, Z. Chen, J. Li, and A. V. Vasilakos, "Preventing distributed denial-of-service flooding attacks with dynamic path identifiers," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1801–1815, Aug. 2017.
[37] N. Moustafa, K. R. Choo, I. Radwan, and S. Camtepe, "Outlier Dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 1975–1987, Aug. 2019.
[38] Z. Jiang, D. Crookes, B. D. Green, Y. Zhao, H. Ma, L. Li, S. Zhang, D. Tao, and H. Zhou, "Context-aware mouse behavior recognition using hidden Markov models," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1133–1148, Mar. 2019.
[39] S. Iranmanesh, F. S. Abkenar, A. Jamalipour, and R. Raad, "A heuristic distributed scheme to detect falsification of mobility patterns in Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 719–727, Jan. 2022.
[40] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.
[41] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.
[42] M. A. Albahar, "Recurrent neural network model based on a new regularization technique for real-time intrusion detection in SDN environments," *Secur. Commun. Netw.*, vol. 2019, pp. 1–9, Nov. 2019.
[43] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.
[44] G. Goswami, N. Ratha, A. Agarwal, R. Singh, and M. Vatsa, "Unravelling robustness of deep learning based face recognition against adversarial attacks," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
[45] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.
[46] X. Ding, Y. Guo, G. Ding, and J. Han, "ACNet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1911–1920.
[47] H. Wang and W. Li, "DDosTC: A transformer-based network attack detection hybrid mechanism in SDN," *Sensors*, vol. 21, no. 15, p. 5047, Jul. 2021.
[48] H. Wang, D. Sahoo, C. Liu, K. Shu, P. Achananuparp, E.-P. Lim, and S. C. H. Hoi, "Cross-modal food retrieval: Learning a joint embedding of food images and recipes with semantic consistency and attention mechanism," *IEEE Trans. Multimedia*, vol. 24, pp. 2515–2525, 2022.
[49] X. Gao, Z. Zhang, T. Mu, X. Zhang, C. Cui, and M. Wang, "Self-attention driven adversarial similarity learning network," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107331.
[50] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. Interspeech*, Sep. 2019, pp. 1–5.

[51] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1963–1971, Mar. 2020.
[52] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," *Int. J. Eng. Technol.*, vol. 7, no. 3, pp. 479–482, Dec. 2018.
[53] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.
[54] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9042–9053, Oct. 2019.
[55] Y. Lin, J. Wang, Y. Tu, L. Chen, and Z. Dou, "Time-related network intrusion detection model: A deep learning method," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

ZIHAN WU received the M.S. degree in applied psychology from the University of Electronic Science and Technology of China, in 2016. He is currently pursuing the M.S. degree in cyber space security with the School of Cyber Security and Computer, Hebei University. His current research interests include network security, intrusion detection, and machine learning.

HONG ZHANG received the Ph.D. degree in computer science from the University of Central Florida, in 2018. He is currently an Assistant Professor with the School of Cyber Security and Computer, Hebei University. His research interests include the design and analysis of parallel systems for big-data computing, which includes two aspects, such as design and analysis. For design, he is currently working on optimizing performance, scalability, resilience, load balancing of data-intensive computing, and distributed machine learning. For the aspect of analysis, he focuses on using program analysis to detect programming errors and performance defects in large-scale parallel computing systems.

PENGHAI WANG is currently pursuing the master's degree with the School of Cyber Security and Computer, Hebei University, Baoding, China. His current research interests include cloud computing, smart communication, machine learning, and the Internet of Things.

ZHIBO SUN received the Ph.D. degree from Arizona State University. He is currently a Cyber Security Researcher. His research interests include human-centric security and threat intelligence analytics.