MFVT:An Anomaly Traffic Detection Method Merging Feature Fusion Network and Vision Transformer Architecture

Ming Li
Shanghai Maritime University https://orcid.org/0000-0001-5290-7446
Dezhi Han (dezhihan88@sina.com)
Shanghai Maritime University https://orcid.org/0000-0001-8861-5461
Dun Li
Shanghai Maritime University
Han Liu
Shanghai Maritime University
Chin- Chen Chang
Feng Chia University

Research

Keywords: Network Intrusion Detection, Traffic Features, Deep Learning, Feature Fusion Network, Vision Transformer, MFVT, CRP, Detection Accuracy

Posted Date: September 17th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-877144/v1

MFVT:An Anomaly Traffic Detection Method Merging Feature Fusion Network and Vision Transformer architecture

First Ming Li[1†], Second Dezhi Han[1*†], Third Dun Li[1†], Fourth Han Liu[1†] and Fifth Chin-Chen Chang[2†]

[1*]Shanghai Maritime University, Shanghai, 201306, China.
[2]Feng Chia University, Taichung, Taiwan.

*Corresponding author(s). E-mail(s): dezhihan88@sina.com;
Contributing authors: 202030310118@stu.shmtu.edu.cn;
446838775@qq.com; liuhanshmtu@163.com;
†These authors contributed equally to this work.

Abstract

Network intrusion detection, which takes the extraction and analysis of network traffic features as the main method, plays a vital role in network security protection. The current network traffic feature extraction and analysis for network intrusion detection mostly uses deep learning algorithms. Currently, deep learning requires a lot of training resources, and have weak processing capabilities for imbalanced data sets. In this paper, a deep learning model (MFVT) based on feature fusion network and Vision Transformer architecture is proposed, to which improves the processing ability of imbalanced data sets and reduces the sample data resources needed for training. Besides, to improve the traditional raw traffic features extraction methods, a new raw traffic features extraction method (CRP) is proposed, the CPR uses PCA algorithm to reduce all the processed digital traffic features to the specified dimension. On the IDS 2017 dataset and the IDS 2012 dataset, the ablation experiments show that the performance of the proposed MFVT model is significantly better than other network intrusion detection models, and the detection accuracy can reach the state-of-the-art level. And, When MFVT model is combined with CRP algorithm, the detection accuracy is further improved to 99.99%.

Keywords: Network Intrusion Detection, Traffic Features, Deep Learning, Feature Fusion Network, Vision Transformer, MFVT, CRP, Detection Accuracy

1 Introduction

The rapid development of the mobile Internet not only brings great convenience to network users and society but also allows criminals to create a series of attacks in the network. These attacks have seriously threatened the normal operation of the network, not only caused a lot of economic losses but also brought hidden dangers to national security. A group of behaviors that violate computer security policies such as confidentiality, integrity, and availability are defined as intrusion detection[1]. As a security protection system used to monitor computer network, the intrusion detection system can detect suspicious behaviors and take corresponding measures to ensure the normal operation of the network and reduce economic losses, which has been in use since the 1980s[2, 3]. Recently, due to the rapid development of mobile Internet, attacks on Internet-connected devices are gradually increasing. Thus, many scholars have a strong interest in the research of intrusion detection systems and good detection results have been achieved[4].

Besides, the detection of anomaly network traffic is an important task of network intrusion detection, which is essential to classify network traffics[5], which requires researchers to make accurate judgments on the collected network traffic data and detect network traffic with offensive behavior. To detect anomaly traffics more effectively, network traffic packets are usually divided into flows according to source IP, destination IP, source port, destination port, protocol and timestamp[6]. The current anomaly traffic detection technology mainly includes: traditional network anomaly traffic detection technology and network anomaly traffic detection method based on machine learning. In this paper, deep learning methods were used to classify network traffics. Deep learning methods have the characteristics of end-to-end and automatic extraction of network traffic data features, to avoid the cumbersome process of manual extraction of features, and deep learning methods have good adaptability, self-organization and promotion ability. So, the use of deep learning can make the detection system have more stable performance and higher detection efficiency[7].

However, deep learning technology needs a large amount of labeled data for training, and labeled data requires experts with specific knowledge to spend a lot of time on labeling, which is time-consuming and laborious. Most of the data sets used in deep learning are imbalanced data sets. These problems cause a significant impact on the performance of deep learning models. Under-sampling and over-sampling are commonly used to solve data imbalance problems, but under-sampling will discard some data leading to the loss of some features, and over-sampling will add some data leading to changing the

original data distribution, both of which have an impact on the experimental accuracy[8]. In this paper, the traffic features learned from a two-layer convolutional networks are fused, which can alleviate the impact of data imbalance on the accuracy of the experiment. Due to the outstanding performance of Transformer architecture in the field of natural language processing (NLP) and the limitations of its application in computer vision, Dosovitskiy [9] improved the Transformer architecture and proposed Vision Transformer architecture for image sequence converter realize image classification and achieved good results. Meanwhile, experiments proved that Vision Transformer required fewer training resources. Inspired by the vision transformer architecture, a deep learning model (MFVT) based on the feature fusion network and the Vision Transformer architecture was proposed in this paper for network anomaly traffic detection. MFVT model has strong ability to deal with imbalanced data sets, and therefore effectively reduce the sample resources required for training. This paper also studies the influence of learning rate change and the number of training epochs on the experimental accuracy based on the MFVT model.

So far, there are many ways to process raw network traffic data, but there is no uniform standard. Since the data that a neural network can accept must be of the same dimension, the extracted network traffic data must be filtered to a specific dimension before it can be used as the input of the neural network model. Most of the traditional methods directly intercept the data of specific dimensions from the network traffic data. Although the effect is quite good, there is room for improvement. Therefore, PCA algorithm is used in this paper to reduce all the processed digital traffic features to a specified dimension. The experimental accuracy obtained in the data sets IDS 2017 [10] and IDS 2012 [11] is significantly higher than the traditional methods.

In summary, the main contributions of this paper are as follows.

- (1)A deep learning model (MFVT) based on feature fusion network and Vision Transformer architecture is proposed, which can effectively improve the detection accuracy while reducing the training resources. On the IDS 2017 dataset and the IDS 2012 dataset, MFVT model can achieve the best performance on all evaluation metrics.
- (2)A new raw traffic data extraction algorithm (CRP) is proposed, which uses the PCA [12] algorithm to reduce the processed digital traffic features to a specified dimension. The ablation experiment results show that the detection accuracy has significantly improved to compare with traditional methods.
- (3)Based on the MFVT model, the impact of training epochs and the variation of the learning rate on the detection performance of the model is further studied.

The rest of this paper is organized as follows. Section 2 introduces the related works to the model and method presented in this paper, Section 3 details the deep learning model and the raw network traffic data processing

algorithm, Section 4 introduces ablation experiments and experimental results of MFVT model in detail, and finally, our work is summarized in Section 5.

2 Related Work

This section mainly summarizes some documents related to the work of this paper, including intrusion detection and Transformer architecture.

2.1 Intrusion detection

In 1980, Anderson [13] proposed the concept of intrusion detection technology, which aims to timely identify abnormal behaviors in the network and reduce losses caused by abnormal behaviors. Over the past 40 years, many methods have been used in intrusion detection, all of which aim to sense attacks with good predictive accuracy and improve real-time prediction. These methods all attempting to extract a pattern from network traffics to distinguish attack traffics from regular traffics.

Specifically, table 1 briefly summarizes the methods used in intrusion detection. Currently, the traditional machine learning methods applied to the field of intrusion detection are mainly supervised learning, such as support vector machine (SVM)[14–16], K-nearest neighbor (KNN)[17], random forest (RF) [18, 19], and so on. These methods mentioned above have a high false alarm rate and a low detection rate for attack traffics. It is a common problem in traditional machine learning methods to design a feature set that can accurately reflect traffic characteristics, and the quality of feature set directly affects the classification performance of the method. In recent years, although many researchers have been working on the problem of how to design feature sets [20, 21], how to design a set of suitable traffic feature sets is still an unresolved research topic.

Moreover, deep learning [22] have good self-adaptability, self-organization, and generalization capabilities. Therefore, it can be a good solution to the problem that traditional machine learning needs to manually design a group of feature sets. The use of deep learning can enable detection systems with higher detection efficiency, and therefore has been widely studied by scholars in recent years. Yan [23] constructed an intrusion detection system based on convolutional neural network (CNN) and applied generative adversarial network to synthesize attack traces, and experimental results verified the effectiveness of the system. Zhang [24] proposed a deep hierarchical network-based intrusion detection model that combines CNN and long short-term memory network (CNN_LSTM), and the CNN_LSTM model achieved good performance on the IDS2017 dataset. Lin [25] constructed a dynamic network anomaly detection system, which uses long and short-term memory network (LSTM) combined with attention mechanism to detect anomalies. Zhang [26] proposed a two-layer parallel learning cross-fusion deep learning model (PCCN), which uses feature fusion technology to improve the extraction of features from small sample data, and experiments on ablation experiments showed good performance.

Table 1: A brief summary of intrusion detection methods.

Author	Method		DataSets	References
Yin C L	Machine Learning	SVM	NSL-KDD	[14]
Reddy R.R			KDD99	[16]
Li W		KNN	Flooding Attack	[17]
Farnaaz N		RF	NSL-KDD Data Set	[18]
Zhang J			KDD99	[19]
Yan Q	Deep Learning	CNN	KDDCUP'99	[23]
Zhang Y		CNN_LSTM	CICIDS2017	[24]
Lin P		Attention+LSTM	CSE-CIC-IDS2018	[25]
Zhang Y		PCCN	CICIDS2017	[26]
Zhong Y		HELAD	KDDCUP99 +CICIDS2017	[27]

Zhong [27] proposed HELAD, a network anomaly traffic detection algorithm integrating multiple deep learning techniques. Although HELAD has better adaptability and detection accuracy, its bit error rate is slightly higher.

2.2 Transformer Architecture

In 2018, Transformer architecture [28] was first appeared in the field of natural language processing (NLP), and it has occupied an important position in the field of NLP. Transformer architecture has been continuously improved by subsequent scholars [29]. Vaswani [30] first constructed Transformer architecture based on attention mechanism. Devlin et al.[31] proposed BERT, a new language representation model, which pretrains a Transformer from unmarked text through joint adjustments of left and right contexts. BERT got the latest results from 11 natural language processing tasks at the time.

Influenced by the excellent performance of Transformer architecture in NLP task, scholars began to extend Transformer architecture to the field of computer vision and achieved good results. Chen et al. [32] constructed a sequence Transformer to perform regression prediction of pixels and obtained competitive results in the image classification task. In 2020, Dosovitskiy et al.[33] proposed a vision Transformer architecture, which uses a pure Transformer to directly extract the features of image block sequences and obtain the most advanced performance on multiple image recognition reference data sets. Besides the most basic image classification tasks, Transformer models are gradually applied to various computer vision tasks, and the number of vision models based on Transformer architecture has gradually become more and more.

In this paper, the latest intrusion detection model based on feature fusion is improved and integrated into Vision Transformer architecture, and then a deep learning model (MFVT) that combines feature fusion network with Vision Transformer architecture is proposed for network anomaly traffic detection. The MFVT takes full advantage of the respective strengths of feature fusion and Vision Transformer architecture, and further improves the detection accuracy of abnormal network traffic by combining with the CPR algorithm proposed by us.

3 Model and methods

This section mainly introduces the CPR algorithm and MFVT model.

In order to improve the processing capacity of existing deep learning models for imbalanced data sets and reduce the required training set resources, in this paper, a new model MFVT and a new raw data processing algorithm CPR were designed. This section mainly introduces the MFVT model and the CPR algorithm. The MFVT model can improve the detection ability of small sample data sets and reduce the training set resources, and the CPR can effectively remove the interference features in the raw data. Figure 1 shows the entire detection process. The MFVT model mainly composed of a feature fusion network and the Vision Transformer architecture. MFVT can use the raw features of network traffics to automatically learn the differences between different categories of network traffic features to classify anomaly network traffics, but the network model requires that the dimensionality of all input data must be consistent, so an algorithm named CPR was proposed to extract the raw features of network traffics and intercept the same dimensional data.

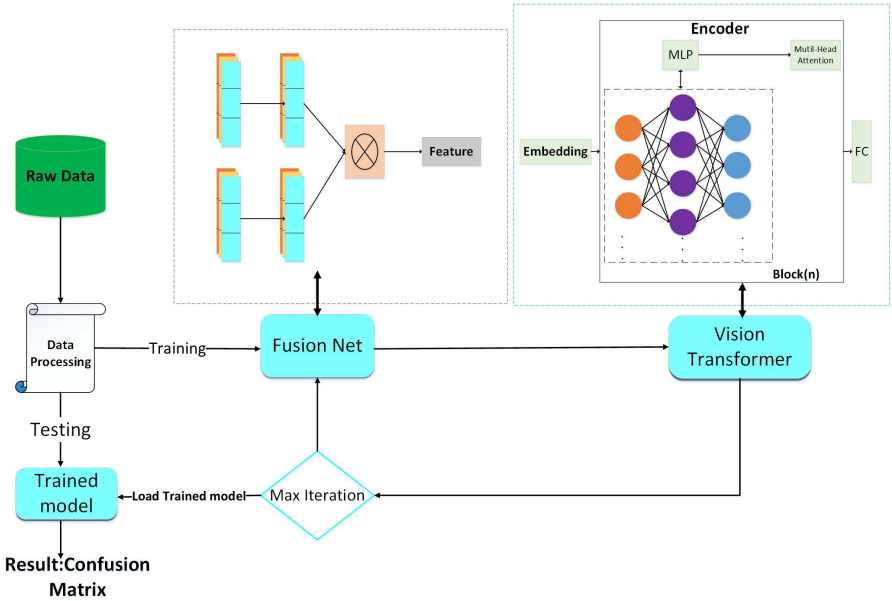


Fig. 1: Anomaly network traffic detection process

3.1 Data processing

The raw data processing algorithm (CPR) proposed in this paper mainly accomplishes the task of extracting raw traffic data from pcap files and processing them into the two-dimensional matrix that required by the network model. Figure 2 shows the entire data processing process.

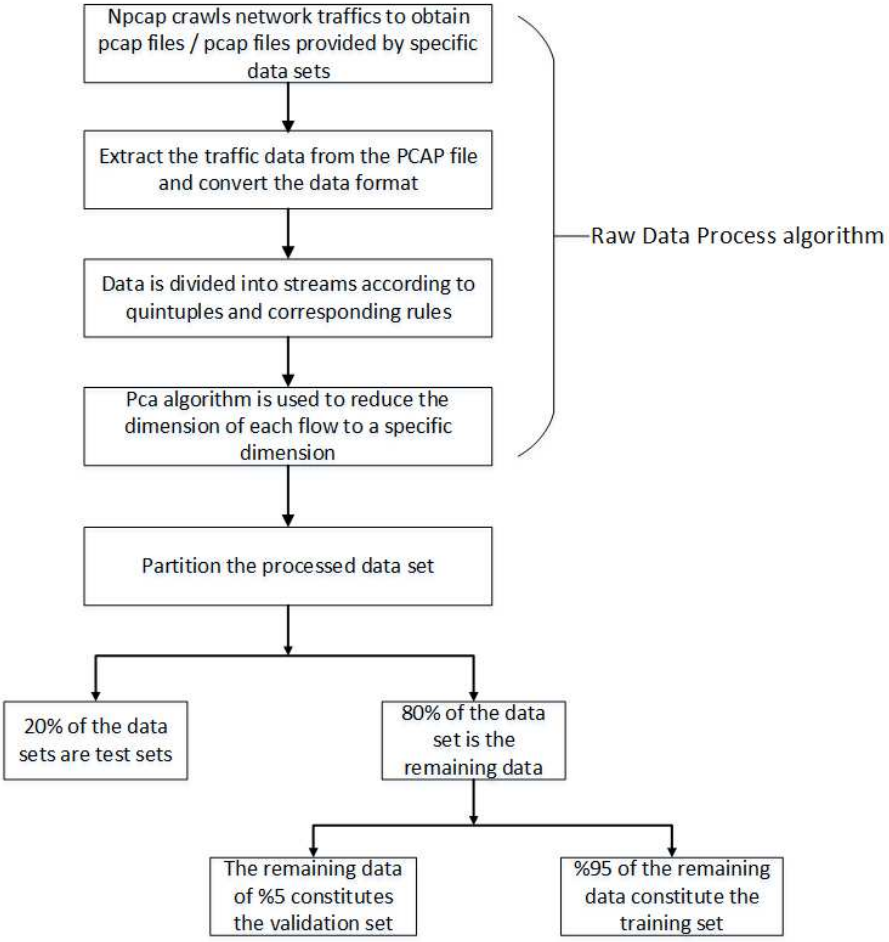


Fig. 2: Overall flow of data processing.

Three steps are required to process the raw flow data into a two-dimensional matrix. The specific steps are as follows.

The first step is to extract the raw data of network traffic from the pcap file, and then convert the extracted byte type data into binary type data.

In the second step, the converted packets are divided into flows according to the five-tuple, and the number of packets and bytes contained in each packet are limited when dividing the flow. If the number of data packets is insufficient, fill in the preceding item, and if the number of bytes contained in the data packet is insufficient, fill in 0. For the completion of this step, refer to the paper [34]. Through the above operations, a data set with fixed dimensions can be obtained. The pseudo code is shown in algorithm 1.

Algorithm 1 Raw data processing

Input: Raw data (pcap);
Output: all_data[];

```
for each pcap do
    if the same five-tuple could be found in the attack Labels then;# Extracting and tagging malicious traffic from pcap files
        Save data and tags into a pcap file
    end if
end for
for each pcap do
        set file name,count=0
    if five-tuple equal and count¿ threshold then;# The maximum number of packets per stream set
        get flows based on five-tuple information of traffic packages
          for each flow do
        Transform flow pcap file into txt file with wirehark to get flow's original hexadecimal data
          end for
    end if
end for
ls=os.listdir (fpath)# The folder path corresponding to each type of attack traffic
for path in ls do
        file=open(path,'r')# Open the txt file where each flow is stored
    for line in file.readlines() do # Read data line by line
        Convert the read data into hexadecimal data and store it in mid_data[] #mid_data[] is to store each packet data in each flow
    end for
        all_data.append(mid_data)
        all_data=[]
end for
```

In the third step, the network traffic data obtained after the first two steps contain high data dimensions and may have redundant features that are useless for network training, which need to be further extracted. In this paper, the data obtained from the first two steps are directly fed into the PCA algorithm

to obtain the data of the required dimensions, and then the data are processed into a two-dimensional matrix. The pseudo code is shown in algorithm 2.

Algorithm 2 Crop and reduce data dimensionality

Input: all_data[];
Output: Data required by neural network;

```
data=[] # Store the final processed data
size # The set number of packets to be intercepted per flow
length # The maximum number of bytes to be intercepted per packet set
for  i=0 to all_data.length do
    if len(all_data[i])¿size then;
     Save data and tags into a pcap file;
        for j to length do
            if all_data[i][j]=="" then; # The number of bytes contained in
the packet is less than the number of bytes intercepted fill in 0
            mid_data.append(0)
             else
            mid_data.append(all_data[i][j])
     data.append(mid_data)
            end if
     Same as above len(all_data[i])¿size
            for i to (size-len(all_data[i])) do
     data.append(The data extracted from the previous)
            end for
        end for
    end if
end forfdata=pca(data, dimension) # Reducing data to a specified
dimension using the pca algorithm
Maxmin_ Normalized(data)
Save the descended data to the specified csv file
```

The main idea of PCA is to map the N-dimensional features to the K-dimension, which is a new orthogonal feature, also known as the principal component, and is a reconstructed K-dimensional feature based on the original N-dimensional features, as shown in Formula 1,2,3,4,5.

$$x_{ij} = x_{ij} - \frac{\sum_{i=1}^{n} x_{i,j}}{n} \quad 0 < i < n, 0 < j < d \tag{1}$$

$$C = \frac{1}{m} X X^T \tag{2}$$

$$w, b = \mathrm{eig}(c) \tag{3}$$

$$p = \text{select}(\text{sort}(w, b), k) \tag{4}$$

$$Y = PX_{n,d} \tag{5}$$

Formula 1 indicates that the original data X is arranged into a matrix with n rows and D columns, and then the matrix is zero-averaged. x_{ij} represents the data in row i and column j of matrix X. In formula 2, c represents the covariance matrix of matrix X. Formula 3 expresses getting the eigenvalue and eigenvector of the covariance matrix c, $eig()$ is the function of getting the eigenvalue and eigenvector, w indicates the obtained eigenvector, and b indicates the corresponding eigenvalue. In formula 4, the eigenvectors are arranged into a matrix in rows from top to bottom according to the corresponding eigenvalues. The first k rows are taken to form the matrix p, where $sort()$ is the sorting function and $slect()$ is the selection function. Formula 5 represents the data set Y obtained after dimension reduction.

3.2 The structure of MFVT

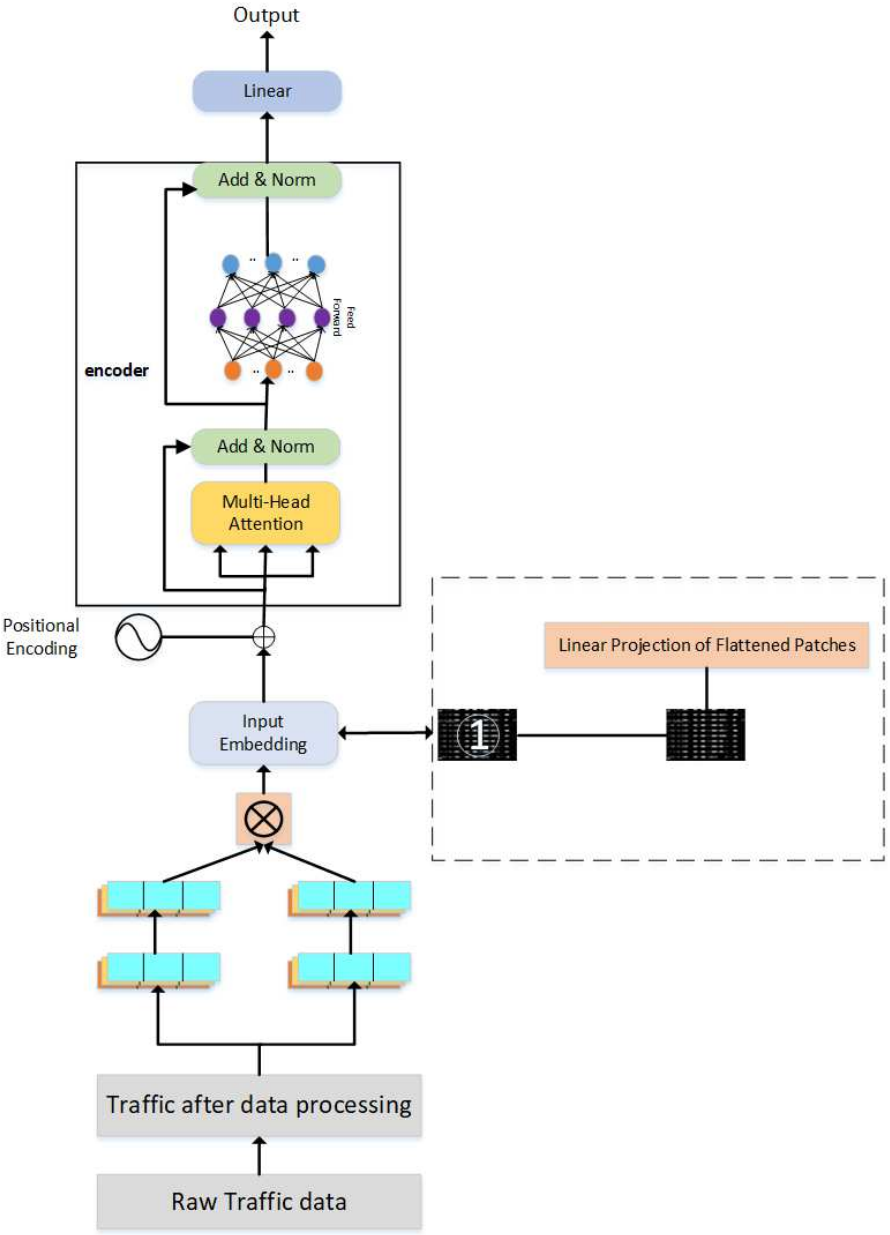


Fig. 3: MFVT's overall structure.

Figure. 3 is the overall structure of the MFVT model, which composed of two parts.

First part is the feature fusion network, which is composed of two layers of parallel convolution networks. The first layer is stacked with two convolution layers, the first convolution has a step of 1, the second convolution has a step of 2, and the size of the kernel is 3. The second layer consists of a convolutional layer and a pooling layer, where the convolutional layer has a kernel size of 3 and a step size of 1, and the pooling layer has a step size of 2. The padding size used in the two-layer convolution process is all 1. To make full use of the features extracted by convolution layer and pooling layer, the extracted features are fused to improve the extraction effect of features for small sample data.The whole calculation process of the feature fusion network is shown in Formula 6,7,8,9,10,11,12,13,14,15,16. Formula 6 represents the padding operation, and formula 7 represents the size change of the output matrix of convolution processing after the padding operation. Under the premise that padding_n is equal to 1, the stride=1 keeps the output size unchanged, and the stride=2 halves the output size.

$$
\begin{aligned}
X = \text{ Padding } (X_0, 1) &= \begin{bmatrix} x_{11} & \cdots & x_{1W} \\ \vdots & \ddots & \vdots \\ x_{H1} & \cdots & x_{HW} \end{bmatrix} \\
&\Rightarrow \begin{bmatrix} 0 & \cdots & 0 & \\ & x_{11} & \cdots & x_{1W} & \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ & x_{H1} & \ldots & x_{HW} & \\ 0 & \cdots & 0 & & \end{bmatrix}
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
H &= \left\lceil \frac{H + padding\_n - w + 1}{\text{stride}} \right\rceil \\
W &= \left\lceil \frac{W + padding\_n - w + 1}{\text{stride}} \right\rceil
\end{aligned} \tag{7}
$$

$X_O$represents the matrix data obtained after the original traffic data is processed by the CPR algorithm. Since the size of the input matrix will be changed after the convolution operation, the padding operation is required to keep the size of the matrix unchanged. X represents the matrix after the padding operation,X_{ij} represents the specific data value in the matrix. W is the width of the matrix, and H is the height.

Formulas 6, 8,9, 10 represent the entire calculation process of the first layer in the feature fusion network. Formulas (3) and (5) represent the convolution operation, V represents the convolution kernel matrix, v_{ij} represents the specific value in the convolution kernel matrix, and k represents the kernel sizes. X_1^1 represents the eigenmatrix obtained after the first convolution operation. Since the stride in formula (7) is 1, the output size remains unchanged. X_1^2

represents the matrix obtained after the padding operation of X_1^1, and X_1^3 represents the eigenmatrix obtained after the second convolution, and the output size is halved because the stride in formula 7 is 2.

$$X_1^1 = X \odot V = \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{k1} & \cdots & x_{kk} \end{bmatrix} \odot \begin{bmatrix} v_{11} & \cdots & v_{1k} \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{kk} \end{bmatrix} \tag{8}$$

$$X_1^2 = \text{ padding } \left(X_1^1, 1\right) \tag{9}$$

$$X_1^3 = X_1^2 \odot V = \begin{bmatrix} x_{11}^2 & \cdots & x_{1k}^2 \\ \vdots & \ddots & \vdots \\ x_{k1}^2 & \cdots & x_{kk}^2 \end{bmatrix} \odot \begin{bmatrix} v_{11} & \cdots & v_{1k} \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{kk} \end{bmatrix} \tag{10}$$

Formulas 6, 11, 12 represent the entire computational process of the second layer in the feature fusion network, whereX_2^1 denotes the feature matrix extracted after the convolution operation, the stride=1 does not change the output size, and X_2^2 denotes the feature matrix obtained after the maximum pooling operation, which halves the size of the output feature matrix.

$$X_2^1 = X \odot V = \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{k1} & \cdots & x_{kk} \end{bmatrix} \odot \begin{bmatrix} v_{11} & \cdots & v_{1k} \\ \vdots & \ddots & \vdots \\ v_{k1} & \cdots & v_{kk} \end{bmatrix} \tag{11}$$

$$X_2^2 = \text{ Maxpooling } \left(X_2^1\right) = \frac{\max\left\{x_{ij}^1\right\}}{i, j \in [1, k]} \tag{12}$$

Formula 13 shows the scale changes of the features extracted from the first and second layers of the feature fusion network. Formula 14 represents the specific process of fusing the first layer with the second layer features. The fusion refers to the summation of the number of channels, but the data must be kept consistent except for the number of channels. C represents the number of channels, $C(1)$ represents the number of channels is 1, $C(32)$ represents the number of channels is 32 and so on, X_f represents the features extracted by the feature fusion network.

$$(C(1), H, W) \Rightarrow \left(C(32), \frac{H}{2}, \frac{W}{2}\right) \tag{13}$$

$$\begin{aligned} X_f &= \left(C(32), \frac{H}{2}, \frac{W}{2}\right) \oplus \left(C(32), \frac{H}{2}, \frac{W}{2}\right) \\ &= \left(C(32+32), \frac{H}{2}, \frac{W}{2}\right) \end{aligned} \tag{14}$$

The second part is composed of the Vision Transformer architecture. To combine Vision Transformer architecture with feature fusion network, the

structure of Vision Tansformer is modified in this paper. The main methods used include: feature embedding, learnable embedding, and Transformer encoder.

For feature embedding, standard Transformer accepts Sequence of Token embeddings as input. To process the feature X_f learned by the feature fusion network, we reconstructed X_f into a flattened 2D block Sequence X_p. Formula 15 shows the specific process of change.

$$\begin{aligned} X_f &\in R^{C(64)\times\frac{H}{2}\times\frac{W}{2}} \\ X_p &\in R^{N\times\left(P^2C(64)\right)} \\ N &= \frac{\frac{H}{2}\frac{W}{2}}{P^2} \\ X_f &\rightarrow X_p \end{aligned} \tag{15}$$

Learnable embedding, a learnable embedding $z_0^0 = x_{class}$ is preset for the feature block embedding sequence, x_{class} denotes the category vector whose state/feature Z_L^0 at the Transformer encoder output is used as the feature representation y, as shown in Formula 21. Learnable embedding is randomly initialized at the beginning of training and obtained by training.

Transformer encoder, which consists of several blocks, each containing a Multi-Head Attention block and a Multi-Layer Perceptron block (MLP), with normalization applied before each block and residual concatenation applied after each block. Figure 4 shows the structure of Head Attention. Formula 16, 17 show how to get the Multi-Head Attention values by Head Attention. Where W_i^Q,W_i^K,W_i^V and W^O are all weight matrices.

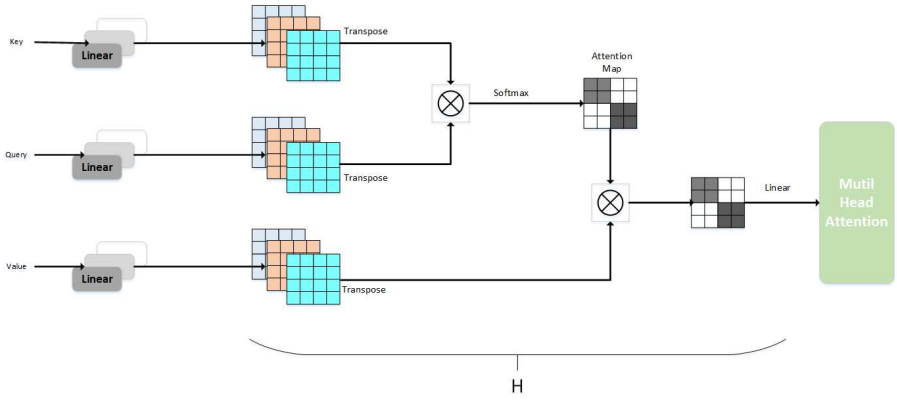


Fig. 4: Attention structure.

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{16}$$

$$\text{Multihead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_b)W^o \tag{17}$$

Table 2: Experimental environment of this paper.

CPU:	i7-10875H CPU@2.30GHz 2.30GHz
RAM:	16G
GPU:	RTX 2060 6G
Compiler Environment:	Python 3.8.2
OS:	Windows 10

Finally, the embedding vectors that combine category vectors and feature block embedding can be input into Transformer Encoder. The Encoder built up by Blocks can extract data features for classification just like CNN. The whole calculation process is shown in formula 18,19,20,21.

$$Z_0 = \left[X_{\text{class}}\,;\, X_P^1 E\right], \quad E \in R^{\left(P^{2\cdot C}\right)\times D}, E_{\text{pos}}\, \epsilon R^{(N+1)\times D} \tag{18}$$

$$Z_l' = MSA\left(\text{LN}\left(Z_{l-}\right)\right) + Z_{l-1}, \quad l = 1 \ldots L \tag{19}$$

$$Z_l = MLP\left(LN\left(Z_l'\right)\right) + Z_l', \quad l = 1 \ldots L \tag{20}$$

$$y = LN\left(Z_L^0\right) \tag{21}$$

The feature embedding block $X_P^1 E$ and the category vector X_{class} form the embedding input vector Z_0. Formula 19 adopts skip connection, where MAS represents Multi-Head Attention operation, LN represents normalization operation, L represents repeatable times, and $Z_l^{'}$ represents the lth output. Formula 20 adopts skip connection, MLP represents the multi-layer perceptron block, L represents repeatable times, and Z_l represents the lth output. y represents the feature representation.

4 Experiments and results analysis

This section first introduces the experimental environment, the datasets IDS 2017 and IDS 2012 used in the experiments, the evaluation criteria used in the experiments, and finally specifies the ablation experiments and some details of the experiments. In the ablation experiments, a series of advanced models were compared with the MFVT model.

4.1 The experimental environment of this paper

In this paper, ablation experiments were conducted on the MFVT model and CPR data processing algorithm under the environment shown in Table. 2.

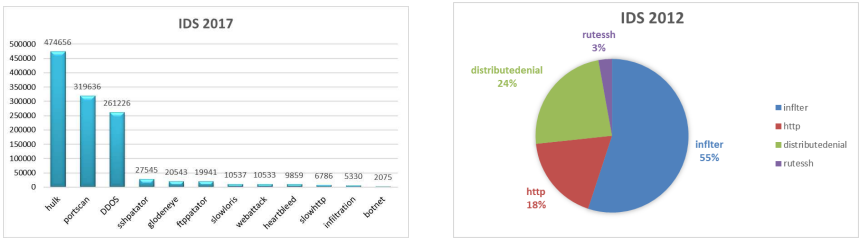
4.2 Datasets

In this paper, A series of ablation experiments were designed using both IDS 2012 and IDS 2017 datasets.

The IDS 2012 dataset contains a week of network activity including both normal and malicious activity, with three days consisting of all normal traffics and the remaining four days consisting of a large amount of normal traffics with a specific type of attack traffics. IDS 2012 dataset contains attack traffic including internal penetration, HTTP denial of service, distributed denial of service using IRC botnet, and brute force cracking of SSH [11].

The IDS 2017 data collection period lasts for five days from 9am on Monday, July 3, 2017 to 5pm on Friday, July 7, 2017, of which Mondays only include normal traffic. The attacks implemented included Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS [10].

Figure 5(a) is a bar chart of the amount of various attack traffics contained in the IDS 2017 dataset, and Figure 5(b) is a pie chart of the amount of various attack traffics contained in the IDS 2012 dataset. It is observed from the figures that both IDS 2017 and IDS 2012 datasets have serious data imbalance problems. The data volume of DDOS, Hulk, and PortScan attacks in the IDS 2017 dataset is significantly larger than that of other types of attacks. The data volume of Inflter attacks in the IDS 2012 dataset directly accounts for 55% of the dataset



(a) (b)

Fig. 5: Percentage of various types of traffic data

4.3 Evaluation Metrics

Authoritative evaluation metrics must be used to judge the merits of a network anomaly traffic detection method. The effectiveness of the machine learning-based network anomaly traffic detection algorithm can be evaluated by the metrics shown in formula. 25,23,22,24. TP represents the positive sample predicted to be positive by the model, which can be called the accuracy rate judged to be true. TN represents the negative sample predicted to be negative by the model, which can be referred to as the percentage of correct judgments that are false. FP represents the negative sample predicted by the model to be positive, which can be referred to as the false alarm rate. FN represents the positive sample predicted to be negative by the model, which can be referred to as the underreporting rate [27].

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

$$F1 - Measure = \frac{2 * \text{ Precision } * \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{24}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

4.4 Ablation experiment and results analysis

In this paper, two datasets of IDS 2012 and IDS 2017 were used for ablation experiments. In addition, this paper also carried out an exploratory study on the impact of model optimization methods on MFVT model detection performance on IDS 2012 dataset.

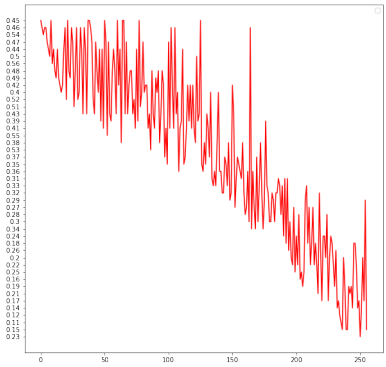
In the MFVT model, the size of the kernels used in the convolutional neural network is $3 * 3$, the segmentation size set in the Vision Transformer architecture is $11 * 11$, the number of the head in the Multi-head attention is 12, and the number of blocks in the Encoder is 12. In the process of model training, the data input batch used in this paper is 256, the epoch of the training iteration is set to 100, and the stochastic gradient descent (SGD) optimizer is used to accelerate the network convergence. The momentum is fixed at 0.9, the learning rate is fixed at 3e-2, weight_decay Set to 0, the loss function uses CrossEntropyLoss. All ablation experiments and results will be described in detail below.

4.4.1 Ablation experiment based on IDS 2012

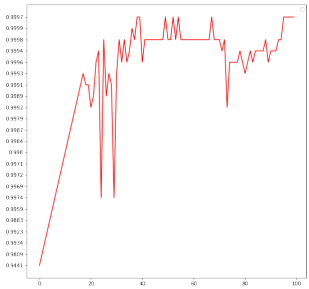
Figure 6 shows the parameter changes of MFVT model when using IDS 2012 dataset for training, including training loss, verification loss and verification accuracy. As show in the picture, the convergence speed of MFVT model is fast, but there are large fluctuations in the later stages of training.

Table 3 shows the experimental results based on IDS 2012 dataset. It is obvious from the table that the MFVT model combined with CPR algorithm proposed in this paper is superior to other methods on all evaluation metrics, reaching the state-of-the-art level. It can also be concluded from the table that MFVT model has superior performance, and its detection accuracy is only slightly worse than that of DT (Decision Tree), but it has higher Precision. To better demonstrate the ability of the MFVT model to deal with imbalanced data, the experimental results of all evaluation metrics of the MFVT model in various types of attack traffic are shown in Table 4.

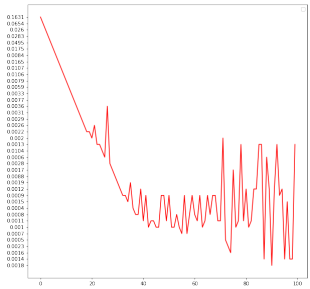
Combining the (B) in Figure 5 and Table. 4 (the experimental results of Infiltrating and Distributedenial, which account for a relatively large proportion, have been marked in red),it can be concluded that the traffic of HTTP and rutesh, which account for a relatively small proportion, still obtains good



(a) Train loss



(b) Val accuracy



(c) Val loss

Fig. 6: Variation of some training results

experimental results. It shows that MFVT model has strong ability to recognize small sample data. The detection performance is further improved by combining the CPR algorithm with the MFVT model.

4.4.2 Ablation experiment based on IDS 2017

The IDS 2012 dataset contains fewer types of attack traffic, and the effectiveness of the MFVT model and the data processing algorithm CPR is demonstrated to be not generalizable on this dataset only. So, ablation experiments also were performed on the more complex IDS 2017 dataset. Figure 7 and 8 are the results of the ablation experiment, from which it can be seen that the accuracy, Recall, F1-score and accuracy of the MFVT model and the combination of MFVT model and CPR algorithm all reached nearly 100%, which was significantly better than other comparison models. Figure 7 shows that the detection results obtained by MFVT model and the combination of MFVT model and CPR algorithm are close to 100% in the evaluation criteria, which is significantly better than other comparison models. The comparison of the FPR between the MFVT model and other comparison models is shown in Figure 8, from which it can be seen that the MFVT model is still the best.

Table 3: Experimental results on the 2012 dataset.

Methods	Precision	Recall	F1-socre	FPR	Accuracy
MFVT	0.9986	0.9975	0.998	0.000525	0.9988
MFVT (CPR)	0.9995	0.9994	0.9995	0.000175	0.9996
Vision Transformer	0.9984	0.9977	0.998	0.000625	0.9985
PCCN	0.9987	0.9979	0.9983	0.000575	0.9986
CNN	0.9958	0.9942	0.9949	0.00145	0.9962
CNN_LSTM	0.9949	0.9936	0.9942	0.001775	0.9951
DLSTM	0.9939	0.9928	0.9933	0.00195	0.9944
KNN	0.993	0.9903	0.9917	0.002125	0.9939
LR	0.9891	0.9902	0.9897	0.00315	0.9909
RF	0.9973	0.9966	0.9969	0.00085	0.9979
DT	0.9984	0.9984	0.9984	0.000375	0.999
SVM	0.9943	0.9937	0.994	0.0018	0.9949

Table 4: Performance of MFVT model and CPR algorithm in each category in IDS 2012 data.

Methods	class	precision	recall	f1-socre	False alarm rate
MFVT	Infiltrating	0.999	0.9994	0.9992	0.0012
	http	0.9984	0.9968	0.9976	0.0004
	distributedenial	0.9985	0.9992	0.9988	0.0005
	rutessh	0.9984	0.9945	0.9965	0
MFVT(CPR)	Infiltrating	0.9995	0.9998	0.9997	0.0006
	http	0.9995	0.9985	0.999	0.0001
	distributedenial	0.9999	0.9999	0.9999	0
	rutessh	0.9992	0.9992	0.9992	0

Combined with the Figure 5(a) and Table 5(experimental results of DDos, Hulk and Portscan, which account for a large proportion of attacks, have been marked in red), it can be concluded that the MFVT model combined with the CPR algorithm has a better ability to recognize small samples.

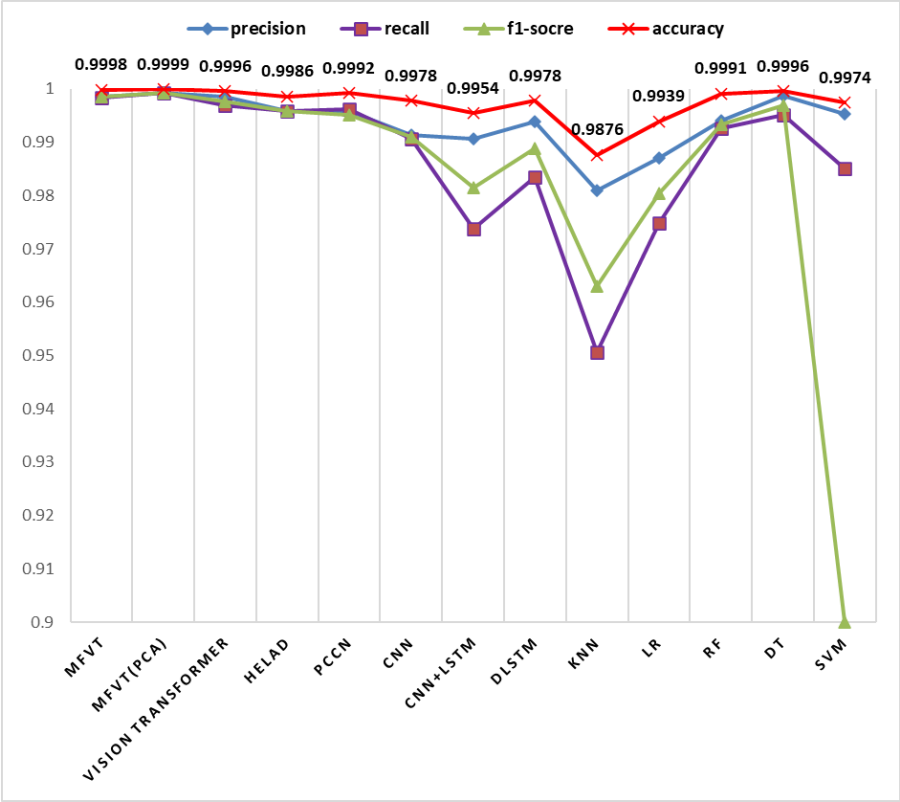


Fig. 7: Partial experimental results1.

Table 5: Performance of MFVT(CPR) in the IDS 2017 dataset.

class	precision	recall	f1-socre
botnet	0.9928	1	0.9964
DDoS	0.9999	0.9999	0.9999
glodeneye	0.9995	0.9998	0.9996
hulk	0.9999	1	0.9999
slowhttp	0.9993	0.9971	0.9982
slowloris	1	0.999	0.9995
ftppatator	1	0.9992	0.9996
heartbleed	1	0.9995	0.9997
infiltration	1	0.9981	0.9991
portscan	1	1	1
sshpatator	0.9996	1	0.9998
webattack	0.9991	0.9991	0.9991

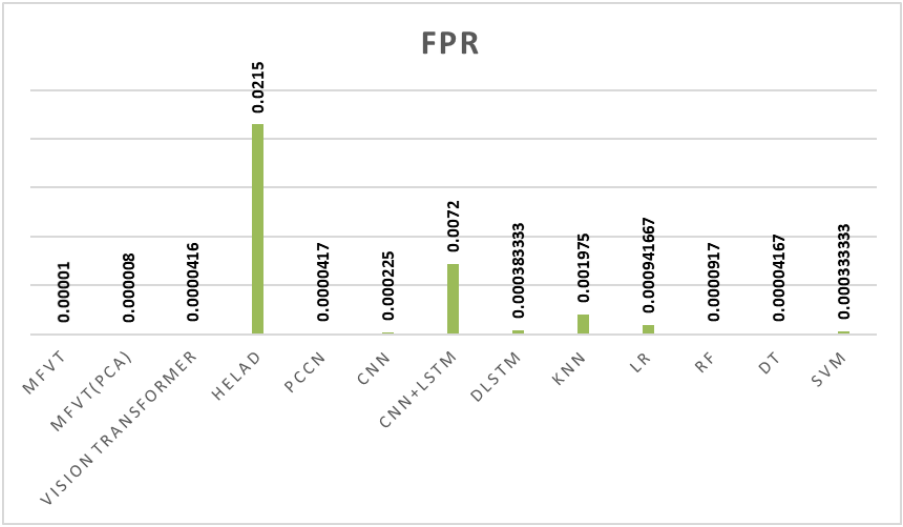


Fig. 8: The result of FPR.

To further demonstrate the error of the prediction results of the MFVT model proposed in this paper combined with CPR, the experimental results were made into the heat map shown in Figure 9. From the heat map, the performance of the MFVT model combined with CPR is very high, and the prediction error rate is extremely low.

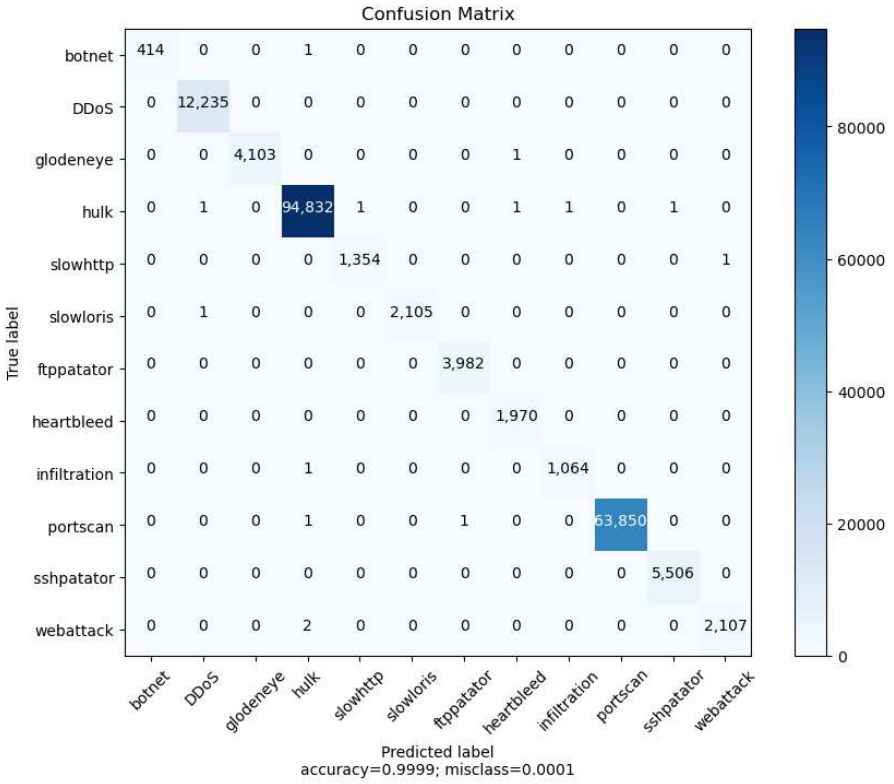


Fig. 9: Heat map of prediction results of MFVT model combined with CPR algorithm.

To verify that the MFVT model can reduce the sample resources required for training, we tested it on the IDS 2017 dataset by reducing the training set data volume according to Formula 26 with all other conditions held constant, $data_0$ is the initial assigned training set data volume, $data_n$ is the updated data volume, and n is taken according to Formula 27, where n_0 is the initial value of n equal to 0.9, and N takes values in the range of 1-7. Table 6 shows the test results.

$$data_n = (1 - 0.1 * n) * data_0 \tag{26}$$

$$n = n_0 - 0.1N \tag{27}$$

Table 6: Test Results-% Of original training data.

Methods	100%	90%	80%	70%	60%	50%	40%	30%
MFVT(CPR)	0.999932	0.999931	0.999929	0.999845	0.999839	0.999783	0.999532	0.999321

As can be seen from Figure 10, when the training set data amount is reduced to 80% of the original training set data amount, the impact on the overall accuracy of the test set is very small. Through this experiment, it is proved that the MFVT model combined with CPR algorithm can effectively reduce the training resources and maintain the accuracy of the test set as much as possible.

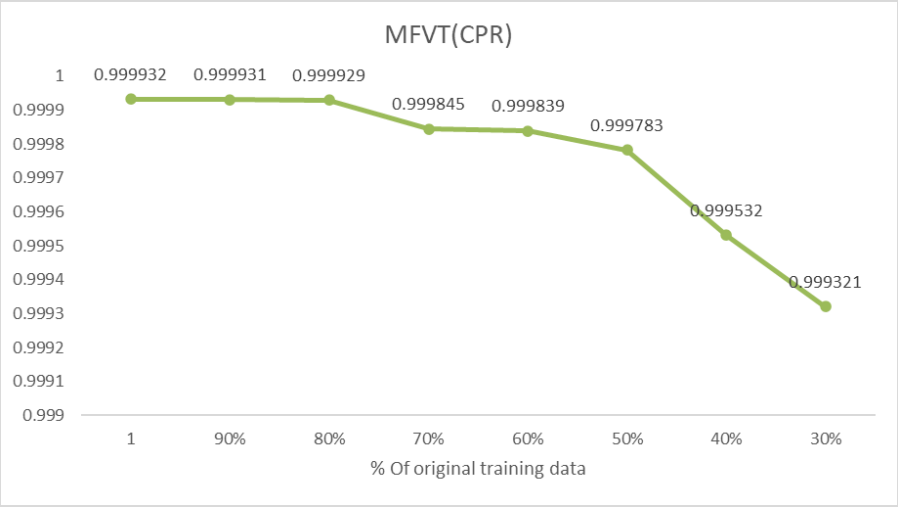


Fig. 10: Experimental results for different training set data amounts.

4.4.3 Optimization of MFVT model

In the conclusion of this section, it is hoped to further improve the detection accuracy and the stability of the model by increasing the training epochs and continuously adjusting the learning rate (lr) during the training process. Thus, IDS 2012 is used as the ablation experiment dataset, which takes less time to train than IDS 2017.Two sets of experiments were conducted. In the first group. our model was trained 1,000 times and the results were recorded every 100 times.

In the second group, based on the first group, lr is changed 100 times per iteration according to formula 28, where lr_i is the learning rate changed every time according to the formula, lr_0 is the initial learning rate, and the epoch is every hundred iterations. To ensure the rigor of the experiment, the values were obtained after conducting the two sets of experiments several times. It can be seen from the figure 11 that both the increase of training epochs and the change of lr can get better prediction accuracy in some intermediate results, but the experimental results tend to be stable in the end. In comparison, the variation of lr will make the variation of experimental results more stable.

$$lr_i = 0.95^{epoch/10} \cdot \mathrm{lr}_0 \tag{28}$$

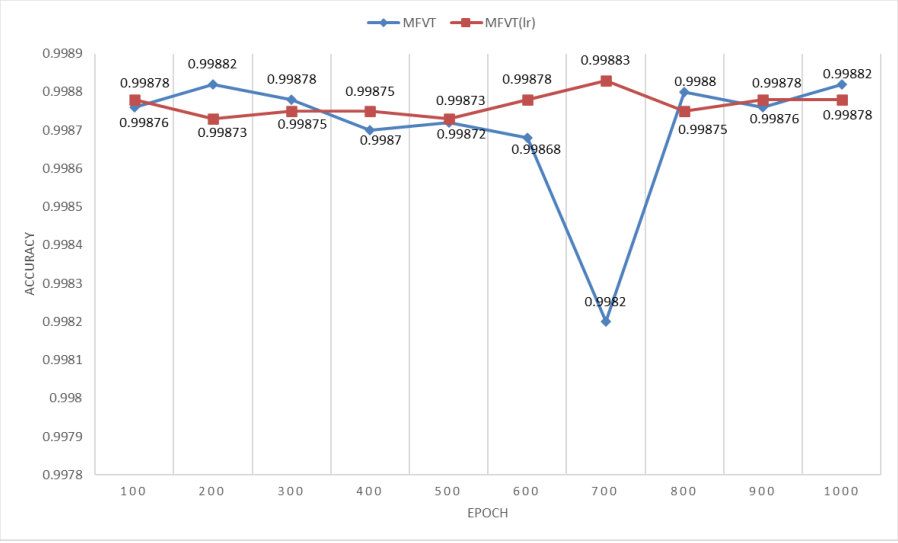


Fig. 11: Experimental results.

5 Results and Discussion

Since most of the deep learning models need a lot of training resources, a network anomaly traffic detection model (MFVT) which combining a feature fusion network with the Vision Transformer architecture was proposed. MFVT can reduce training resources while maintaining high detection accuracy. In this paper, a new raw traffic data extraction algorithm (CRP) was proposed. The MFVT model combined with the CRP algorithm achieved nearly 100% detection accuracy on both datasets IDS 2012 and IDS 2017, and with much better performance than the other methods in the comparison experiments. The MFVT model combined with the CRP algorithm is more capable of handling imbalanced data sets and can further improves the detection accuracy of the experiment.

Although the MFVT model combined with the CRP algorithm has an excellent performance in the field of anomaly traffic detection, the scalability of the model is weak and the detection accuracy of new types of attack traffic that do not appear in the training set needs to be improved in the face of the increasingly complex network environment and the emergence of new attack types.

Considering the importance and practical significance of scalability, the scalability of the MFVT model will be further improved in the future to enhance the practical value and practical significance of the model.

List of Abbreviations

MFVT: An Anomaly Traffic Detection Method Merging Feature Fusion Network and Vision Transformer architecture

CPR: a new raw traffic features extraction method

PCA: Principal Component Analysis

Declarations

Funding

This research is supported by the National Natural Science Foundation of China under Grant 61873160, Grant 61672338 and Natural Science Foundation of Shanghai under Grant 21ZR1426500.

Conflict of interest

There have no conflict of interest

Authors' contributions

All authors read and participated in the manuscript's completion

Authors details

Ming Li is currently pursuing his master's degree at Shanghai Maritime University. His main research interests focus on deep learning and network security.

Dezhi Han received the BS degree from Hefei University of Technology, Hefei, China, the MS degree and PhD degree from Huazhong University of Science and Technology, Wuhan, China. He is currently a professor of computer science and engineering at Shanghai Maritime University. His specific interests include storage architecture, Blockchain technology, cloud computing security and cloud storage security technology.

Dun Li received the M.S. degree from the Macau University of science and technology. He is currently doing the Ph.D. degree in the Shanghai Maritime University. His main research interests include smart finance, big data, machine learning, IoT, and Blockchain

Han Liu received the M.S. degree from the Shanghai Maritime University, where he is currently pursuing the Ph.D. degree. His main research interests include big data, cloud computing, distributed computing, cloud security, machine learning, IoT, and blockchain.

Chin-Chen Chang received the Ph.D. degree in computer engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1982, and the B.E. and M.E. degrees in applied mathematics, computer and decision sciences from National Tsinghua University, Hsinchu, Taiwan, in 1977 and 1979, respectively. He was with National Chung Cheng University, Minxiong, Taiwan . Currently,

he is a Chair Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan, since 2005. His current research interests include database design, computer cryptography, image compression, and data structures. Prof. Chang was a recipient of many research awards and honorary positions by and in prestigious organizations both nationally and internationally, such as the Outstanding Talent in Information Sciences of Taiwan. He is currently a Fellow of the IEEE, a Fellow of the IEE, U.K and a Member of the IEICE.

References

[1] Hung-Jen, L., Chun-Hung, R.L., Ying-Chih, L., Kuang-Yuan, T.: Intrusion detection system:a comprehensive review. Journal of Network & Computer Applications. **36**(1), 16–24 (2013)

[2] Weller-Fahy, D.J., Borghetti, B.J., Sodemann, A.A.: A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Communications Surveys & Tutorials **17**(1), 70–91 (2015)

[3] Ajith, A., Crina, G., Carlos, M.V.: Evolutionary design of intrusion detection programs. International Journal of Network Security **4**(3) (2007)

[4] Anwar, S., Mohamad Zain, J., Zolkipli, M., Inayat, Z., Khan, S., Anthony Jnr, B., Chang, V.: From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions. Algorithms **2017** (2017)

[5] Ajith, A., Crina, G., Carlos, M.V.: A survey of network anomaly detection techniques. Journal of Network and Computer Applications **60**(3), 19–31 (2016)

[6] Zhang, J., Chao, C., Yang, X., Zhou, W., Yong, X.: Internet traffic classification by aggregating correlated naive bayes predictions. IEEE Transactions on Information Forensics & Security **8**(1), 5–15 (2013)

[7] Zhang, Y., Chen, X., Jin, L., Wang, X., Guo, D.: Network intrusion detection: Based on deep hierarchical network and original flow data. IEEE Access **7**, 37004–37016 (2019)

[8] Oksuz, K., Cam, B.C., Kalkan, S., Akbas, E.: Imbalance problems in object detection: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**(99), 1–1 (2020)

[9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2020)

[10] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: International Conference on Information Systems Security & Privacy (2018)

[11] Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security **31**(3), 357–374 (2012)

[12] Smith, L.I.: A tutorial on principal components analysis. Information Fusion **51**, 52 (2002)

[13] Anderson, J.P.: Computer security threat monitoring and surveillance (1980)

[14] Yin, C.L., Zhu, Y.F., Fei, J.L., He, X.Z.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access **PP**(99), 1–1 (2017)

[15] Kuang, F., Xu, W., Zhang, S.: A novel hybrid kpca and svm with ga model for intrusion detection. Applied Soft Computing **18**(C), 178–184 (2014)

[16] Reddy, R.R., Ramadevi, Y., Sunitha, K.: Effective discriminant function for intrusion detection using svm. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2016)

[17] Li W, W.Y.e.a. Yi P: A new intrusion detection system based on knn classification algorithm in wireless sensor network. Journal of Electrical & Computer Engineering

[18] Farnaaz, N., Jabbar, M.A.: Random forest modeling for network intrusion detection system. Procedia Computer Science **89**, 213–217 (2016)

[19] Random-forests-based network intrusion detection systems. IEEE Transactions on Systems Man & Cybernetics Part C **38**(5), 649–659 (2008)

[20] Dhote, Y., Agrawal, S., Deen, A.J.: A survey on feature selection techniques for internet traffic classification. In: International Conference on Computational Intelligence & Communication Networks (2016)

[21] Zhang, H., Lu, G., Qassrawi, M.T., Zhang, Y., Yu, X.: Feature selection for optimizing traffic classification. Computer Communications **35**(12), 1457–1471 (2012)

[22] Imagenet classification with deep convolutional neural networks. In: NIPS (2012)

[23] Yan, Q., Wang, M., Huang, W., Luo, X., Yu, F.R.: Automatically synthesizing dos attack traces using generative adversarial networks. International journal of machine learning and cybernetics **10**(12), 3387–3396 (2019)

[24] Zhang, Y., Chen, X., Jin, L., Wang, X., Guo, D.: Network intrusion detection: Based on deep hierarchical network and original flow data. IEEE Access **7**, 37004–37016 (2019)

[25] Lin, P., Ye, K., Xu, C.Z.: Dynamic network anomaly detection system by using deep learning techniques. International Conference on Cloud Computing (2019)

[26] Zhang, Y., Chen, X., Guo, D., Song, M., Wang, X.: Pccn: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows. IEEE Access **PP**(99), 1–1 (2019)

[27] Zhong, Y., Chen, W., Wang, Z., Chen, Y., Li, K.: Helad: A novel network anomaly detection model based on heterogeneous ensemble learning. Computer Networks, 107049 (2019)

[28] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)

[29] Han, K., Wang, Y., Chen, H., Chen, X., Tao, D.: A survey on visual transformer. Computer Vision and Pattern Recognition (2020)

[30] Radford, A.: Language models are unsupervised multitask learners (2019)

[31] Kim, M., Kim, G., Lee, S.W., Ha, J.W.: St-bert: Cross-modal language model pre-training for end-to-end spoken language understanding. ICASSP (2020)

[32] Chang, Y., Huang, Z., Shen, Q.: The same size dilated attention network for keypoint detection. In: Artificial Neural Networks and Machine Learning -ICANN 2019: Theoretical Neural Computation (2019)

[33] Chung, J., Gulcehre, C., Cho, K.H., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. Eprint Arxiv (2014)

[34] Ming Li, X.Y.e.a. Dezhi Han: Design and implementation of an anomaly network traffic detection model integrating temporal and spatial features. Security and Communication Networks **2021** (2021)

Figure Title and Legend

- Figure 1: Anomaly network traffic detection process
- Figure 2: Overall flow of data processing
- Figure 3: MFVT's overall structure
- Figure 4: Attention structure.
- Figure 5: Percentage of various types of traffic data
- Figure 6: Variation of some training results , (a)Train loss ,(b) Val accyracy ,(c)Val loss
- Figure 7: Partial experimental results , Blue represents precision, purple represents recall, green represents F1-score, and red represents accuracy
- Figure 8: The result of FPR
- Figure 9: Heat map of prediction results of MFVT model combined with CPR algorithm.
- Figure 10: Experimental results for different training set data amounts.
- Figure 11: Experimental results. Blue represents MFVT, red represents accuracy MFVT change lr.