IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 9, 2019, accepted October 30, 2019, date of publication November 6, 2019, date of current version November 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951839

An Intrusion Detection System Based on a Quantitative Model of Interaction Mode Between Ports

AO LIU AND BIN SUN

Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin 300384, China

Corresponding author: Bin Sun (sunbin1024@126.com)

This work was supported in part by the National Natural Science Foundation under Grant 71501141.

ABSTRACT Considering the characteristics of network traffic on the data link layer, such as massive high-speed data flow, information camouflaged easily, and the phenomenon that abnormal traffic is much smaller than the normal one, an intrusion detection system (IDS) based on the quantitative model of interaction mode between ports is proposed. The model gives the quantitative expression of Port Interaction Mode in Data Link Layer (PIMDL), focusing on improving the accuracy and efficiency of the intrusion detection by taking the arrival time distribution of traffic. The feasibility of the model proposed is proved by the phase space reconstruction and visualization method. According to the characteristics of long and short sessions, a neural network based on CNN and LSTM is designed to mine the differences between normal and abnormal models. On this basis, an improved Intrusion Detection algorithm based on a multi-model scoring mechanism is designed to classify sessions in model space. And the experiments show that the quantitative model and the improved algorithm proposed can not only effectively avoid camouflage identity information, but also improve computational efficiency, as well as increase the accuracy of small sample anomaly detection.

INDEX TERMS Anomaly detection, interaction mode between ports, intrusion detection, neural network, phase space reconstruction.

I. INTRODUCTION

To avoid the serious losses caused by network attacks, it is important to build an effective intrusion detection model to explore the existing characteristic rules in mass traffic data. As a branch of machine learning, deep learning can recognize the internal law of a certain kind of things to the maximum through training multilayer neural network, so it has a unique advantage to explore the internal law of abnormal attack traffic in massive network traffic data.

Among the many problems involved in intrusion detection, the anomaly detection method is the most important one, and its key point is to design a feature set that can accurately describe network traffic [1], [2]. At present, many data sets, such as KDD'99 [3], NSL-KDD [4], UNSW-NB15 [5], CIC-IDS-2017 [6], ISCX [7], which are widely used in intrusion detection systems, have a large capacity and rich characteristics, and the neural network can be used to mine the internal rules of these data sets to realize the intrusion detection. There are a lot of achievements in previous studies, while ignoring several problems. Firstly, to obtain the previous feature set from the initial traffic, it is necessary to check all the traffic data in the first two seconds and the first 100 connections at the end of the session, however, the intrusion detection system cannot be too complex because of the massive and high-speed traffic characteristics, in practice, according to previous research methods, building feature sets from the real-time generated initial traffic will cause a lot of computational burdens. Secondly, previous studies have trained neural networks based on a large number of high-level protocol information (e.g. logon status, flag). When attackers camouflage these attributes, the classification accuracy of neural networks will be greatly affected. Thirdly, in reality, the scale of abnormal traffic is usually smaller than that of normal traffic, but the abnormal traffic of data sets used in previous research accounts for a large proportion. For example, in the training set provided by NSL-KDD, the abnormal traffic accounts for 46.52% of the total traffic,

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran Tariq.

but it is almost impossible for the abnormal traffic of this scale to appear in reality (by examining 16.7+ billion visits to 100,000 randomly-selected domains on the Incapsula network, IMPERVA's 2018 attack traffic report said abnormal traffic accounts for 21.8% of total traffic, and in some specific scenarios, such as small local area networks, the potential threat is less, and the proportion of abnormal traffic will be smaller.).

To solve the problems above, this paper proposes PIMDL, which reconstructs the traffic feature set from the initial traffic to quantify the network traffic. Consider the arrival time distribution of data packets as time series, the traditional training method based on high-dimensional traffic feature set (e.g. KDD'99) is improved. The main work is as follows:

- A detailed definition and quantitative representation of PIMDL is proposed, and PIMDL is visualized using phase space reconstruction to show the difference between normal traffic and abnormal traffic, thus proving its feasibility.
- Autocorrelation function (ACF) is used to explore the characteristics of long and short sessions, respectively, and the richness and bias are considered to determine the range of sample selection. The details of the construction of the neural network are determined by the above methods.
- Three neural networks based on Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) are used to mine the differences of PIMDL between normal and abnormal, (the specific parameters are shown in Table 2, 3, 4).
- Design a multi-model scoring mechanism to evaluate network traffic, map sessions into three-dimensional model space, use Support Vector Machine (SVM) to classify session traffic in model space, and finally implement traffic intrusion detection. This algorithm is the core of our work, its theoretical basis is based on the research before section C of Chapter IV. The algorithm describes the specific process of continuously acquiring traffic data packets (frames) for intrusion detection in the process of traffic generation. The effectiveness of the improved algorithm is proved by the final design comparison experiment.

II. RELATED WORKS

Previous studies have provided a wide range of ideas for the combination of deep learning and intrusion detection. Khan *et al.* [8] proposed a two-level deep learning model, Two-Stage, which integrates the two decision-making stages into the detection process. In the first stage, normal and anomaly are divided, and attack types are defined in the second stage. Yin *et al.* [9] proposed an intrusion detection scheme using recurrent neural networks (RNN), which improved the accuracy of multi-task classification. Javaid *et al.* [10] proposed a STL-based intrusion detection model, which uses two groups of deep confidence echo networks to fully extract the characteristics of attack traffic and complete the high-precision detection of intrusion traffic. Vinayakumar *et al.* [11] proposed a model migration approach, scale-hybrid-IDS-AlertNet (SHIA), which uses KDD'99 data set to train the full connection layer, and tests it on NSL-KDD, UNSW-NB15 data sets. It improves the accuracy of model migration between different data sets. Shone *et al.* [12] proposed a method of dimension reduction using NDAE instead of DBN, which is more suitable for dimension reduction of asymmetric dimension data. Wang *et al.* [13] proposed a method of intrusion traffic classification based on representation learning, which classifies traffic according to different application layer protocols and uses convolutional neural network to learn its behavior projection in link layer. This method distinguishes the different manifestations of different applications in the data link layer. Jiang *et al.* [14] proposed LSTM-RNN multi-channel voting algorithm, which improved the adaptability of neural network in different environments and expanded the application scenarios of detection algorithm. Kolosnjaji *et al.* [15] proposed a traffic intrusion detection method based on convolution and feed-forward neural structure, which realized the extraction of hierarchical features of data sets. Zhou *et al.* [16] summarized the collaborative intrusion detection systems to expand the deficiency of traditional IDS in detecting coordinated attacks. Naseer *et al.* [17] uses a convolution neural network, automatic encoder and cyclic neural network to construct depth network, which can learn the features of existing data sets and improve the accuracy of classification. Wang *et al.* [18] proposed an end-to-end encryption traffic classification method based on a one-dimensional convolutional neural network. This method integrates feature extraction, feature selection and classifier into a unified end-to-end framework, automatically learns the non-linear relationship between the original input and expected output, and improves the detection accuracy of VPN traffic.

TABLE 1. Datasets usage.

KDD' 99	NSL-KDD	UNSW-NB15	ISCX
[8][11][12][13]	[8-12][14][17]	[8][11]	[18]

Table 1 shows the datasets used in the above literature, previous studies have relied too much on existing feature sets, which will lead to the problems described in the introduction. In this paper, the intrusion detection system is designed by using only the initial traffic (PCAP file) in the existing data set without using the feature set.

III. PIMDL MODEL AND ITS CHARACTERISTIC ANALYSIS

A. PIMDL MODEL

Communication under TCP/IP protocol can be regarded as information exchange between ports, the process of information exchange between ports is called the session. In the process of data interaction, session presents a specific interaction mode over time, ideally, this mode strictly abides by

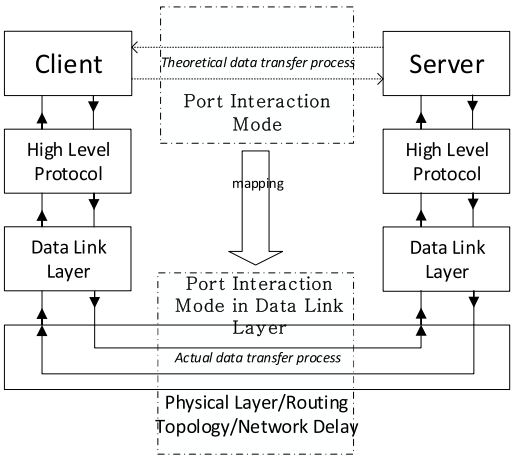


FIGURE 1. Formation mechanism of PIMDL: data flow is shown by arrows. Sessions between ports pass through constraints and are transmitted through data packets at the data link layer, the mode of port interaction can be found at the data link layer.

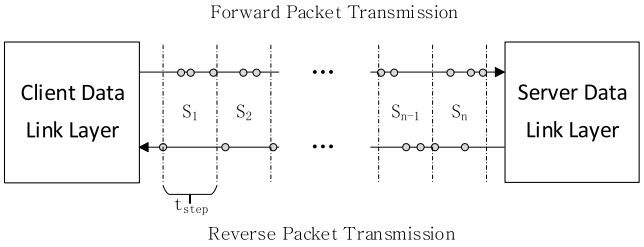


FIGURE 2. Quantitative representation of PIMDL: intercepting statistical information of bidirectional packet distribution based on step Δt.

the TCP/IP protocol, but in reality, because the actual data transmission is transmitted through the data link layer in the form of data packets (or as a frame, the term of data packet pays more attention to the specific description of frame data, which includes IP, port, etc.), the interaction mode is affected by high-level protocols, routing topology, uncontrollable network delay, etc., it will show a more flexible rule in the data link layer. From the point of view of the data link layer, the arrival time distribution of data packets represents the port interaction mode to some extent. The definition of PIMDL is derived from this, its formation mechanism is shown in Figure 1.

In Figure 1, the dotted line frame above is the ideal data transmission channel. Its essence is the session between ports, which follows the TCP/IP mode. However, since the real data stream is shown by the real line, the session will project an interactive mode represented by the arrival distribution of data packets in the link layer in the dashed frame below the graph. This interactive mode is projected by the high-level session in the data link layer, that is, the model proposed in this paper, PIMDL. The process of building PIMDL model based on initial traffic is as follows:

According to IP and port number of both sides in duplex network, network traffic is classified as session set. $C = \{c_1, \ldots, c_n\}$, among them $c_i \in C$ is a session, Let t_{c_i} denote the duration of c_i, the starting time of c_i as the benchmark (zero time), the ending time of t_{c_i} and the step length of Δt, and segment c_i into n_i time periods. $n_i = t_{c_i}/\Delta t$, as shown in Figure 2;

Based on the above segmentation method, c_i data packets are classified into two categories: forward and backward, and the distribution of bidirectional data packets in each time period is counted. Set that the number of forward packets is p_j^+, the average packet interval time is $t_j^+ = \Delta t/p_j^+$, the number of backward packets is p_j^-, and the average packet interval time is $t_j^- = \Delta t/p_j^-, j = 1, \ldots, n_i$. If there is no data transmission between ports in a certain period of time, the above four attributes are 0;

Based on the above method, c_i can be quantified as a ordered feature set $S_i = \{s_1, \ldots, s_{n_i}$, among them, $s_j = (p_j^+, t_j^+, p_j^-, t_j^-)$ is a quaternion, it can be expressed as a four-dimensional time series S_i, which can represent the interaction mode of its high-level ports to a certain extent.

B. FEASIBILITY VERIFICATION OF PIMDL BASED ON PHASE SPACE RECONSTRUCTION

To verify the feasibility of PIMDL model, the embedding theorem [19] is used to visualize the model. On this basis, the difference between normal traffic and abnormal traffic is analyzed to verify whether it is consistent with the reality. Four-dimensional time state series S_i is transformed into matrix by phase space reconstruction of embedding theorem. Think of each element as a pixel and color it $(p_j^+, t_j^+, p_j^-, t_j^-)$ to get a color picture.

Chaotic time series is a kind of irregular movement in determining the system. PIMDL is affected by the high-level network protocol, network topology, and other determinant factors, showing a more complex form of expression. Therefore, it can be approximated as a chaotic time series to a certain extent. Based on the embedding theorem, a phase space is reconstructed from chaotic time series in the same topological sense as the original dynamic system. The process is as follows:

For time series $\{x_i | i = 1, \ldots, n\}$, The corresponding high-dimensional phase space $\{y_i | i = 1, \ldots, n - (dim - 1)lag\}$ can be reconstructed, it can be seen as a matrix. Among them, the *lag* is the bit of delay time and the *dim* is the embedding dimension, $y_i = \{x_i, x_{i+lag}, \ldots, x_{i+(dim-1)lag}\}$. Mutual Information method and False Nearest Neighbor method are used to determine the values of the *lag* and *dim*, respectively.

1) LAG CALCULATING

Optimal *lag* value of time series can be determined by mutual information method, the main process is to divide a period of time series $X = \{x_1, \ldots, x_n\}$ into two successive subseries according to lag, the initial elements of which are x_i and x_{i+lag}, respectively, and to explore the change of the Mutual Information [20] between them with the increase of *lag*. When the mutual information reaches the first few minimum values, it is the optimal value of *lag*.

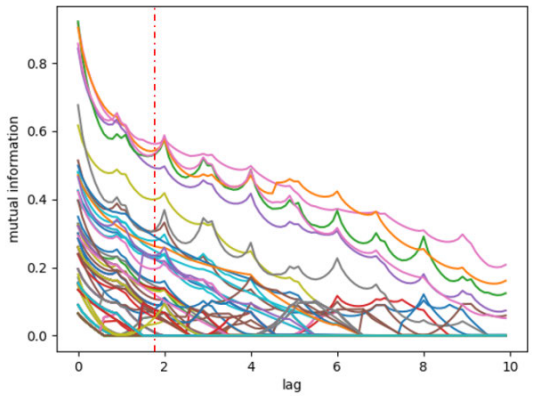


FIGURE 3. Mutual information changes with lag.

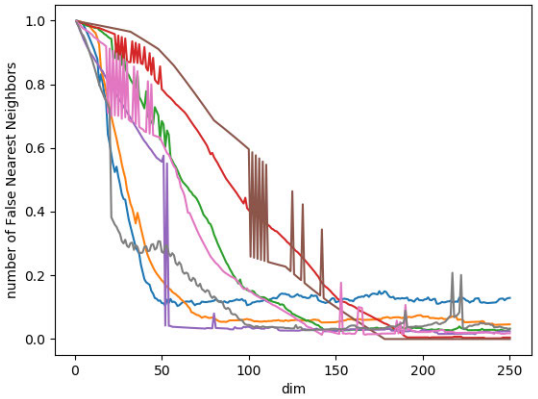


FIGURE 4. FNN changes with dim.

Select MAWI dataset for network traffic from 00:00 to 02:00 in the morning of May 9, 2018 [21]. PIMDL is obtained by $\Delta t = 0.1$s, and its attribute p_j^+ is selected to transform it into one-dimensional time series to test the change of Mutual Information with *lag*. Select sessions with the duration of 500-600 s, and divide the initial traffic data into 120 groups according to the number of packages from 2000 to 8000 with intervals of 50. One sample is randomly sampled from each group for the experiment. The results are shown in Figure 3.

The horizontal axis in Figure 3 represents the *lag* and the vertical axis represents the mutual information between two successive sub-series with corresponding *lag* values. The results show that when the Mutual Information reaches the second minimum, the *lag* of the random sample is concentrated around 1.8s. Therefore, according to the above results, the *lag* selection of phase space reconstruction is 1.8s.

2) DIM CALCULATING

For the chaotic time series, the proportion of False Nearest Neighbors (FNN) [22] is calculated from the minimum value of dim $=$ 2. Gradually increasing the value of dim until the proportion of FNN is less than 5% or the number of FNN no longer decreases with the increase of dim, it can be considered that the chaotic attractor has been fully opened, and take the value of dim in this case as the final result [23]. Select sessions with the duration of 500-600 s, and divide the initial traffic data into 8 groups according to the number of packages from 200 to 120 with intervals of 125. One sample is randomly sampled from each group for the experiment. The results are shown in Figure 4.

As shown in Figure 4, the horizontal axis represents *dim* and the vertical axis represents the number of FNN corresponding to *dim* at lag $= 0.6$. As can be seen from Figure 4, FNN converges almost completely when $dim = 160$, that is to say, it is an ideal case when $dim \geq 160$ is used in the current data set.

3) VISUALIZATION OF PIMDL

In order to verify the feasibility of the proposed PIMDL model, it is visualized to show the difference between normal and abnormal intuitively. For normal traffic and abnormal traffic, sessions with more than 200 packets and duration of 500-600 s are captured, and $\Delta t = 0.6$ s is taken to quantify all sessions into PIMDL (Table 1), to get a collection of PIMDL with all sessions.

Phase space reconstruction for each $S_i = \{s_1, \ldots, s_{n_i}\}$ in PIMDL set, a new state matrix $S_i^{re} = \{s_1^{re}, \ldots, s_{n_i-(dim-1)lag}^{re}\}$ is obtained, in which $s_j^{re} = \{s_j, s_{j+lag} \ldots, s_{j+(dim-1)lag}\}$. S_i^{re} is a matrix of order $[n_i - (dim - 1)lag] \times dim$, each element in the matrix has its corresponding attribute set $s_j = (p_j^+, t_j^+, p_j^-, t_j^-)$, considering the element as a pixel, the corresponding four attributes are scaled to the parameter range $(0, 255)$ of RGBA color format (red, green, blue, opacity) according to the maximum value. To make the image clearer, add 1 at the position of the fourth element t_j^-, that is, converting the original four elements:

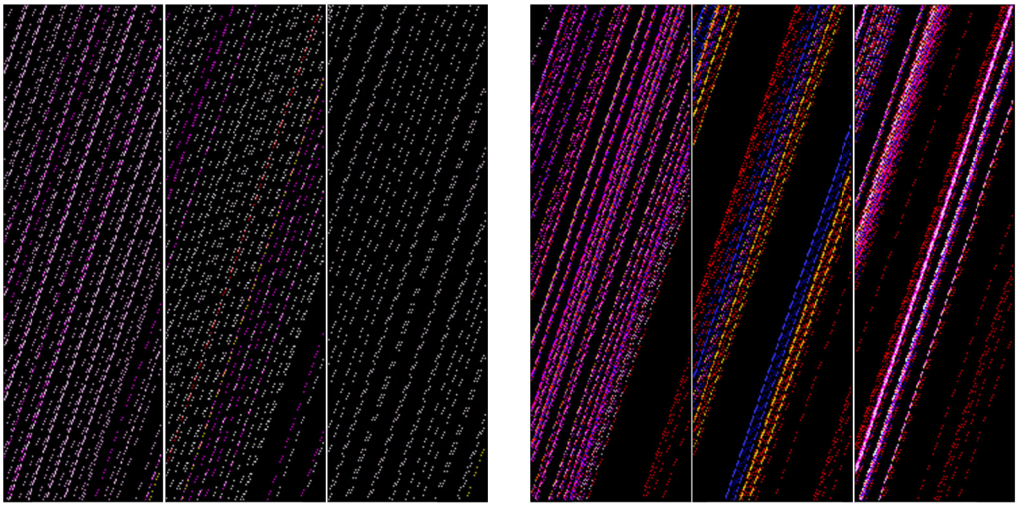
$$s_j = (p_j^+, t_j^+, p_j^-, t_j^-) \tag{1}$$

into:

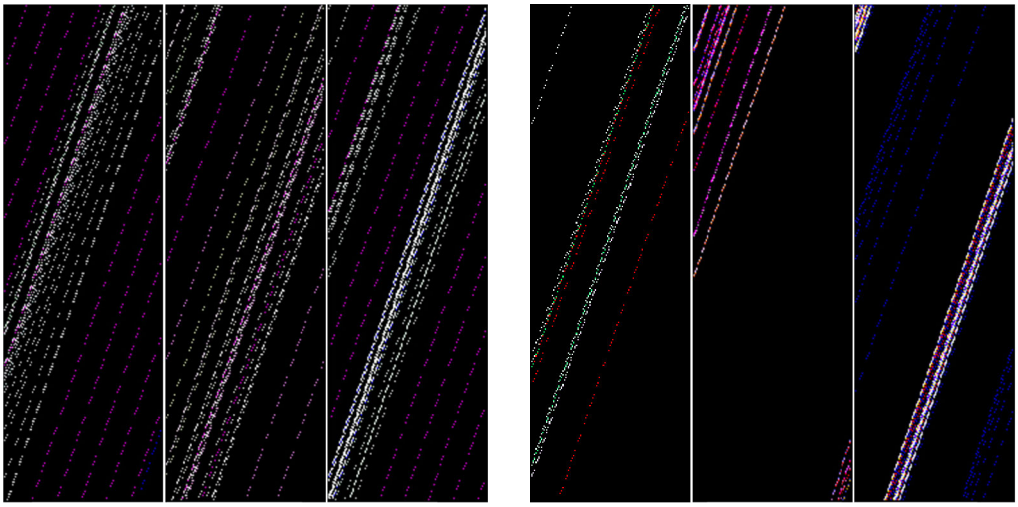
$$s_j^{pixel} = \left(\frac{p_j^+}{p_{j\ max}^+}, \frac{t_j^+}{t_{j\ max}^+}, \frac{p_j^-}{p_{j\ max}^-}, \frac{t_j^-}{t_{j\ max}^-} + 1 \right) \tag{2}$$

among them, $p_{j\ max}^+, t_{j\ max}^+, p_{j\ max}^-, t_{j\ max}^-$ are the maximum values of four attributes in the group, respectively. Finally, the S_i^{re} is transformed into a $[n_i - (dim - 1)lag] \times dim$ color image. According to the conclusions above, take $lag = 1.8$s, $dim = 160$, CICIDS-2017 and CTU-13 [24] datasets are taken as the data sources of anomaly attack traffic. The results are shown in Figure 5

From Figure 5, it can be seen that the color of the normal flow is single (left), while the color of abnormal flow is bright and diversified (right). The experimental results are consistent with the actual situation of the network environment, that is, the normal interaction mode of sessions follow strict TCP/IP protocol and has a consistent interaction mode, while the abnormal mode has no obvious regularity, and maintains a large gap with the normal mode. The above experiments prove the feasibility of using PIMDL to quantify network traffic.



(a) the case of a large number of data packets



(b) the case of a smaller number of data packets

FIGURE 5. Normal & abnormal sessions, each of these rectangular subgraphs represents a session.

C. CONSISTENCY VERIFICATION OF PIMDL BASED ON AUTOCORRELATION FUNCTION

According to the duration of network sessions, traffic can be classified into two categories: longer sessions and shorter sessions. Because the data characteristics determine the selection of the types of neural networks, the self-correlation of time series is explored for two types of sessions, and then the neural networks are constructed according to their respective attributes.

For the PIMDL of each group of sessions, the consistency is verified by using ACF. The ACF is defined as follows [25]

$$R(t_s, t_e) = \frac{cov(X^s, X^e)}{\sigma_s \cdot \sigma_e} \tag{3}$$

Two continuous stochastic processes with t_s and t_e as the first elements and the largest capacity are intercepted from the initial stochastic process X. They are X^s and X^e, respectively, $|X^s| = |X^e| = r.\mu$ and σ denote expectation and standard deviation respectively, $cov()$ denotes covariance, which is defined as follows:

$$cov(X^s, X^e) = \sum_{i=0}^{r} (x_i^e - x_i^t) \tag{4}$$

If the delay time is $\tau = t_e - t_s$ and t_s is taken as the reference time, i.e. 0 time, then formula (1) can be expressed as the following expression

$$R(\tau) = \frac{cov(X^0, X^\tau)}{\sigma_0 \cdot \sigma_\tau} \tag{5}$$

For stochastic processes with fixed length in formula (5), when τ increases, the capacity of X^s and X^e decreases in order to ensure the same and continuous capacity.

In addition, if the values of the ACF of two stochastic processes are in the confidence interval, it is considered that the two stochastic processes are not correlated under a certain confidence level. The definition of the confidence interval of the autocorrelation function is as follows [26].

$$\Delta = 0 \frac{\sqrt{2} \cdot erf^{-1}(acc)}{\sqrt{L}} \tag{6}$$

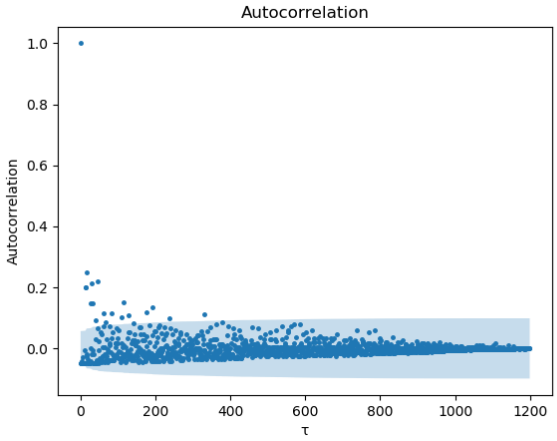


FIGURE 6. Autocorrelation function of shorter session (Similar results were found in other samples).

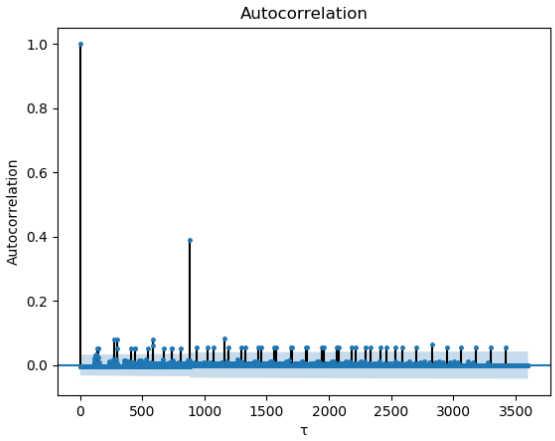


FIGURE 7. Autocorrelation function of longer session. In order to highlight the difference between the two types of sessions, each data point in the autocorrelation function graph of a long sessions has a line with the horizontal axis. (Similar results were found in other samples).

In formula (6), *acc* denotes degree of confidence, L denotes capacity, and $erf()$ denotes error function, which is defined as follows

$$erf(\beta) = \frac{2}{\sqrt{\pi}} \int_0^{\beta} e^{-y^2} dy \tag{7}$$

Based on the above formulas, the autocorrelation function and confidence interval of a time series are plotted with the change of τ. A set of 17806 port sessions from 00:00 to 02:00 of MAWI dataset on May 9 was obtained. According to the PIMDL quantification method, a set of PIMDL with single sample size of 1200×4 are obtained with $\Delta t = 0.01$s for short sessions lasting 11-12s. Because the ACF can only analyze one-dimensional time series, each element of the multivariate time series needs to be calculated separately. Therefore, each sample first selects its first attribute, p_j^+, to make it into 1200×1 one-dimensional time series. The corresponding sample autocorrelation function is shown in Figure 6.

In Figure 6, the horizontal axis represents τ, and the vertical axis represents Pearson correlation of X^s and X^e. The light blue region is the confidence interval of 95% under the corresponding τ. As can be seen from the Figure 6, with the increase of τ, the ''cut off'' appears in the autocorrelation graph, that is, the autocorrelation value of the sequence converges in the confidence interval. The experimental results show that when τ increases to a certain extent, X^s and X^e no longer have obvious correlation, that is, the sequence no longer shows obvious autocorrelation. The time series under this phenomenon is called ''stationary time series'', which indicates that the front and back states of the time series are independent of each other.

Because the attribute t_j^+ of PIMDL is completely dependent on p_j^+, the stability of samples in t_j^+ case is the same as that in p_j^+ case. Using the above method, the time series under p_j^- and t_j^- conditions are also ''stationary time series''.

Compared with shorter sessions, a set of PIMDL with single sample size of $3600\times \sim 4$ are obtained with $\Delta t = 1$s for short sessions lasting 3500-3600s. To avoid image clutter, take the time interval of 1 s and repeat the drawing process of the above autocorrelation graph. The experimental results are shown in Figure 7.

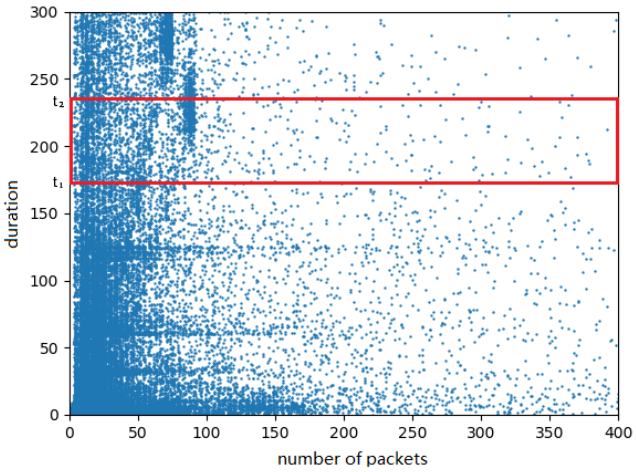


FIGURE 8. Session distribution and window selection.

As can be seen in Figure 7, with the increase of τ, there is no ''cut off'' phenomenon in the series autocorrelation function graph, that is, the series does not show good stationarity. The value of autocorrelation function appears periodically outside the confidence interval, which indicates that a longer session sequence can be approximated to an ''autocorrelation series'', that is, the time series has the properties of front-back correlation. Similar to the above, the samples under other attributes show same properties.

D. SAMPLING RANGE DETERMINATION

When processing samples, the neural network needs to unify the samples into the same shape of tensor for input. Therefore, before training the neural network, it is necessary to ensure that all samples have the same shape. Ideally, all PIMDLs

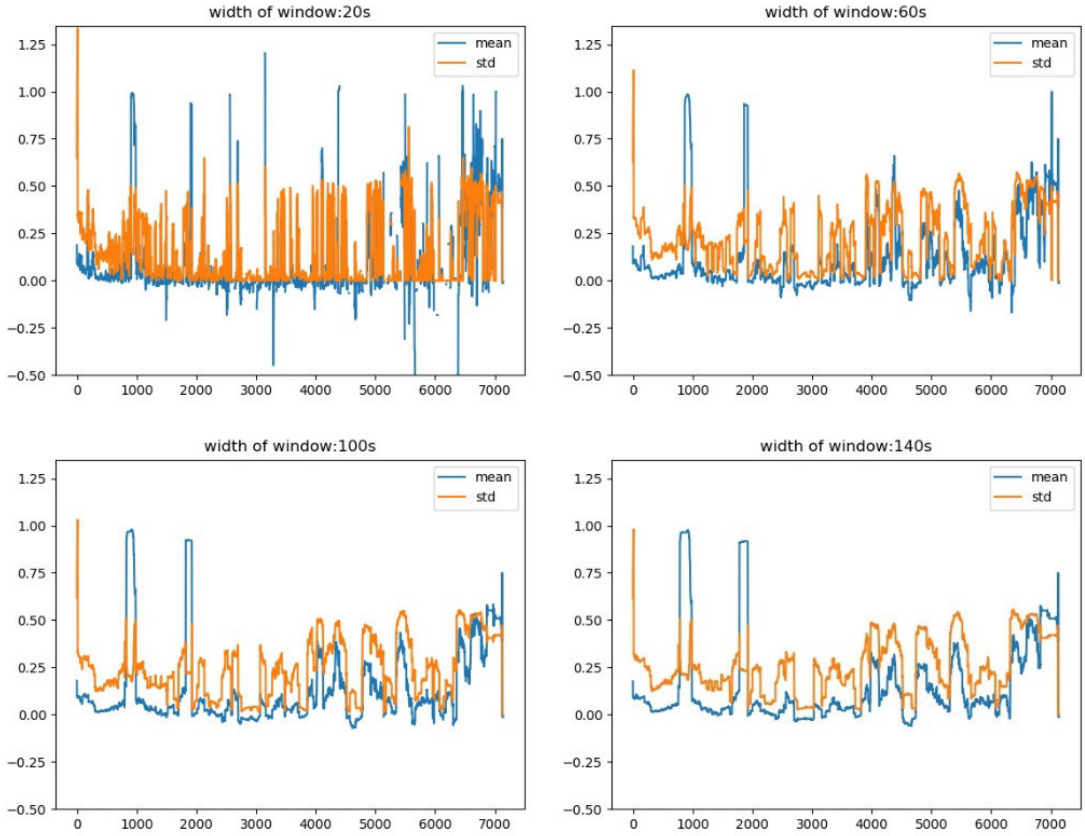


FIGURE 9. In the case of different windows, the variation of std and mean of Pearson correlation sets with window sliding.

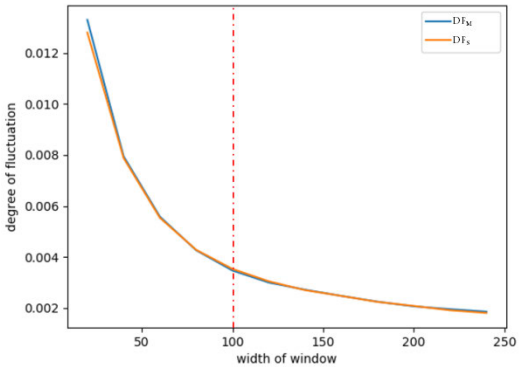


FIGURE 10. Variation of fluctuation intensity with w value.

are input into a set of neural networks for training, because durations of the sessions are different, when acquiring their PIMDL for shorter sessions, 0 is used to fill the missing bits in order to conform to the shape of longer sessions. However, this method will result in a large bias in shorter sessions under unified training. To avoid this bias, the samples are grouped according to the duration of the session, and each group trains different neural networks, as shown in Figure 8.

In Figure 8, the horizontal axis represents the number of packets in the session, and the vertical axis represents the duration of the session, each point represents a session. The sessions lasting from t_1 to t_2 are grouped, and the sessions in the group are complemented into the same shape for input by neural network training. Let the window $w = t_2 - t_1$, with the increase of w, in order to satisfy the input format of the neural network, the number of 0 bits filled gradually increases, which results in the increase of sample richness and bias in the group at the same time. On the contrary, with the decrease of w, the bias will be smaller, but too small range will make the samples in the group unable to summarize all the characteristics of this period group, and it leads to too many trained neural networks, each neural network has the possibility of over-fitting (That is to say, the accuracy of training set is better, but it cannot be generalized to the realistic prediction) because of its small sample capacity. Based on the above considerations, it is necessary to analyze the two aspects of the impact of w changes: richness and bias, so as to determine the optimal selection range. The richness is measured by the sample diversity and the bias is quantified by the difference between sample time and upper limit time in the group.

1) DIVERSITY ANALYSIS OF INTRA-GROUP SAMPLES

Based on the above traffic data, the following experiments are designed to determine the influence of window range on sample richness in the group:

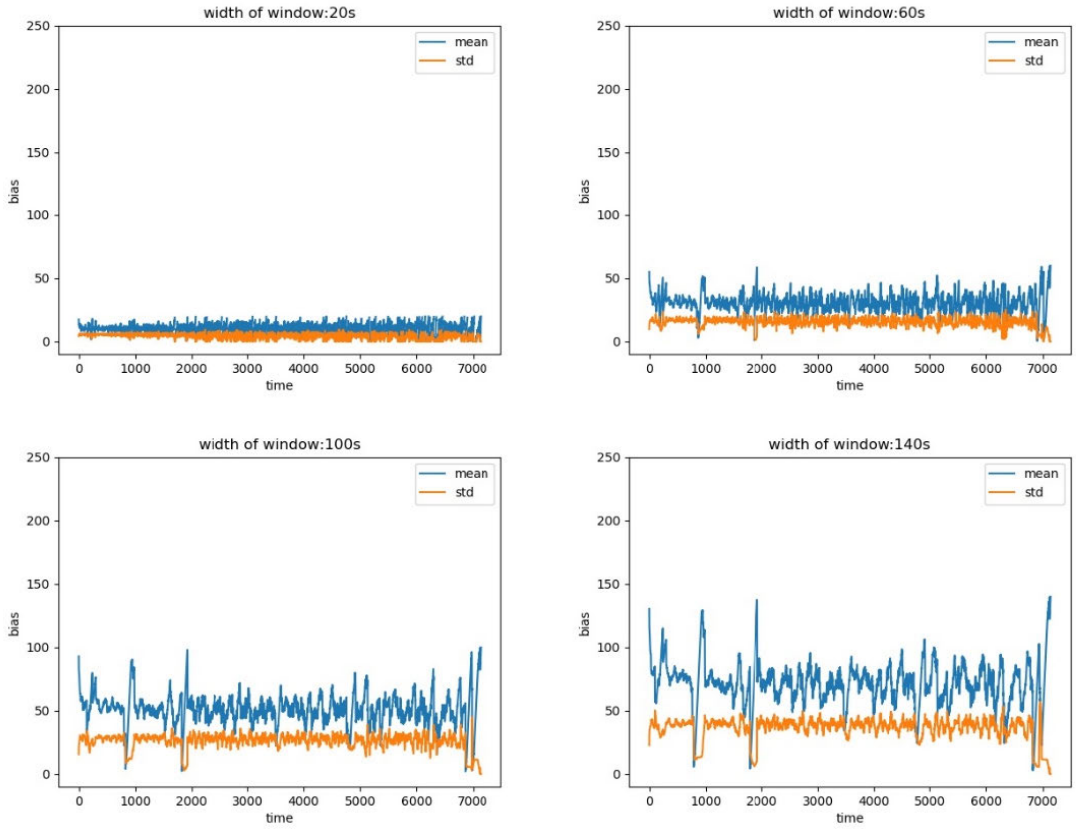


FIGURE 11. In the case of different windows, the variation of std and mean of difference value sets with window sliding.

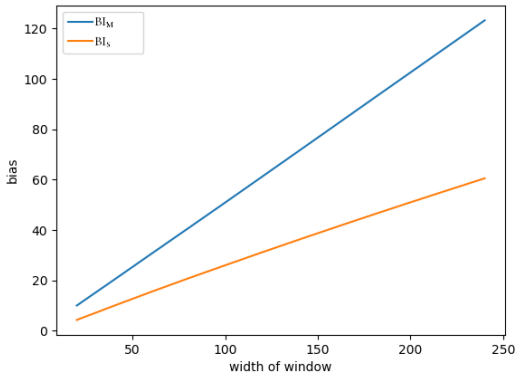


FIGURE 12. Variation of bias with w value.

Define w as window width and slide the window on the time axis. Taking 1s as the sliding window step, the sessions in the window are classified into corresponding time windows, which take [0,w], [1,w+1],..., [7200-w,7200], respectively, and each time window specifies a set of samples;

For each group of samples, the first-order difference of the occurrence time of all session c_i is made, that is, the difference between two adjacent terms is calculated, and a new set of time series is obtained. For all series in the set, Pearson correlation ($\rho_{X,Y} = \frac{cov(X,Y)}{\rho_X \rho_Y}$, X, Y are two independent stochastic processes) between any two is calculated, and then Pearson correlation sets of each group of samples are obtained, which is used to measure the intra-group session diversity of samples.

The standard deviation(std) and mean value of Pearson correlation set corresponding to each sample group are calculated, drawing the changing trend of them along with window sliding on the coordinate system;

Change w from (20, 40, ..., 240) and repeat the above experimental process. The results are shown in Figure 9.

As can be seen, with the window sliding, there is no obvious increasing or decreasing trend in the std and mean of sample differences within the group. When the window exceeds 100s, the image fluctuation tends to be similar. Let the time series of std and mean be M_{diff} and S_{diff}, respectively, and quantify their fluctuation intensity by their total variation degree. That is, $DF_M = \sum |t_{i-1} - t_i|, t_i \in M_{diff}$ and $DF_S = \sum |t_{i-1} - t_i|, t_i \in S_{diff}$, Figure 10 shows how DF_M and DF_S vary with the value of the window w.

The horizontal axis in Figure 10 shows the size of the window w, and the vertical axis is the corresponding values of DF_M and DF_S. It can be seen that the fluctuation intensity shows a slow attenuation state with the increase of the window. When the window exceeds 100s, the fluctuation intensity tends to converge.

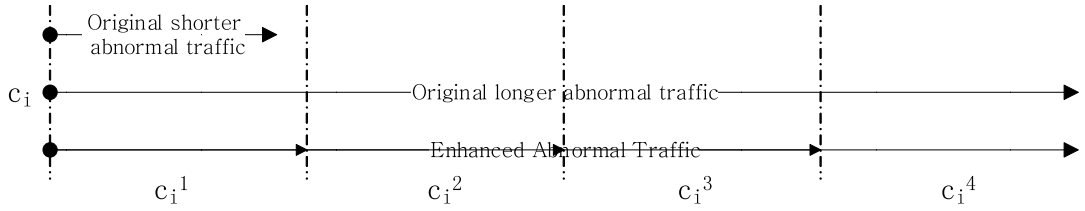


FIGURE 13. Data enhancement method for abnormal traffic.

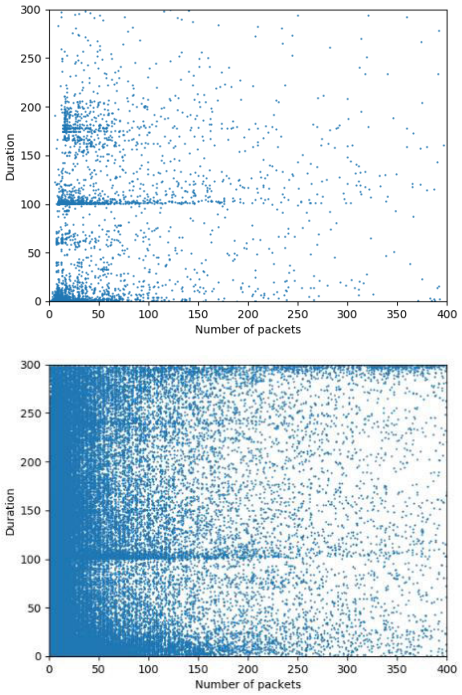


FIGURE 14. Data distribution before and after data enhancement, above and below, respectively.

2) BIAS ANALYSIS OF INTRA-GROUP SAMPLES

Based on the above traffic data, the following experiments are designed to determine the influence of window range on sample bias in the group:

For each sample group, because the bias is mainly determined by the complement position, the difference value set between the upper limit of container duration and all session durations are counted to indicate the session bias in this sample group;

The standard deviation and mean value of the difference value set corresponding to each sample group are calculated, drawing the changing trend of them along with window sliding on the coordinate system;

Change w from (20, 40, …, 240) and repeat the above experimental process. The results are shown in Figure 11.

As can be seen from Fig 11, with the change of window width, there is no obvious increasing or decreasing trend in the standard deviation and mean value of sample differences within the group. Let the time series of std and mean be M_{bias} and S_{bias}, respectively, and quantify their bias by their mean value. That is, $BI_M = mean\,(M_{bias})$ and $BI_S = mean\,(S_{bias})$, Figure 12 shows how BI_M and BI_S vary with the value of the window w.

In Figure 12, the horizontal axis represents the size of the window w, and the vertical axis represents the corresponding bias degree. It can be seen that the bias presents a form of an approximate linear function with the increase of the window.

3) OPTIMAL w SELECTION

In the above two experiments, Fig. 10 shows that the fluctuation intensity decreases slowly with the increase of the window. When the window is 20s, the characteristic diversity of samples in each time window is relatively narrow, and there is almost no significant difference between samples. When the time window slides, diversity fluctuates present violently because the window is too small to accommodate a large number of characteristics. With the increase of the window, the sample characteristics in the time window are more abundant, and the variation of the degree of diversity is gradually smooth. When the window is larger than 100s, the mean value of the fluctuation degree tends to converge and the variation of diversity degree tends to be the same, which means that the richness of characteristics in the window has reached the ideal value. Figure 12 shows that the deviation increases linearly, with the increase of the window, the bias in all samples in the container increases linearly, and there is no obvious inflection point or extreme value. When the window is larger than 100s, the change of bias is approximate when the deviation is less than 100s, that is to say, only considering the bias cannot get a better choice.

Because the ideal state of all samples in the container needs to satisfy the requirements of rich characteristics and small bias at the same time, the window size is selected to be 100s based on the above two experiments. In addition, because the richness and bias of the fixed window have no obvious increasing or decreasing trend, there is no special requirement for selecting the starting position of the window, that is, the optimal range constraint of sample acquisition only considers the size of the window to be 100s.

IV. NEURAL NETWORK AND INTRUSION DETECTION

A. DATA ENHANCING FOR ABNORMAL TRAFFIC

At present, abnormal traffic data sets have little consideration for long-term interaction information between ports, and cannot fully summarize the temporal characteristics of abnormal interaction. The accuracy of deep learning depends greatly on the size of the training set. If input samples cannot

TABLE 2. Parameters of the experiment.

Number of experiments	1	2	3
Number of models	model_1	model_2	model_3
Sample selection range	$t_{c_i} \in (0,100)$	$t_{c_i} \in (100,200)$	$t_{c_i} \in (200,300)$
Δt of PIMDL	0.05	0.1	0.1
Sample capacity(normal, abnormal))	(5601, 4796)	(1520, 3288)	(1604, 4606)
Training epochs	100	300	
Learning rate	0.0002	0.0014	
Batch size	256	512	
K value of K-fold	4		
Data split ratio of train, validate and test	0.6: 0.2: 0.2		
Loss function	Binary crossentropy		
Optimizer	Rmsprop		

TABLE 3. Structure of model_1.

Layer	Type	Filters/neurons	stride	Activation Function	Other attributes
1	Conv_1D	24	6	sigmod	—
2	Max_Pooling	—	4	—	—
3	Flatten	—	—	—	—
4	Dense	128	—	sigmod	L2 (0.01) regularization
5	Dense	1	—	sigmod	—

summarize the characteristics of all samples in this category, it will appear obvious over-fitting. In addition, the existing datasets (such as CICIDS-2017) have a large number of long-term abnormal sessions lasting more than 24 hours, and the features of longer and shorter sessions are different, unified processing is prone to data waste. To solve this problem, based on the separability of the network traffic[27], a data enhancement method of interaction mode between abnormal traffic ports is proposed. A long session is segmented into several short sessions by time series to obtain the characteristics of different time periods. The method is shown in Figure 13.

All data of CICIDS-2017 and CTU-13 were obtained, and 60% of them were randomly sampled as training samples, and the remaining 40% as follow-up simulation test samples. Determine the maximum session time t_{max} of enhancement samples, and enhance the data of exception set C_a : for each $c_i \in C_a$, if it's duration $t_{c_i} < t_{max}$, the data will not be enhanced; otherwise, it will be segmented at t_{max} intervals, c_i will be enhanced $\left\lceil \frac{t_{max}}{t_{c_i}} \right\rceil$ samples, and c_i will be deleted from the data set. Figure 14 shows the effect of data enhancement on abnormal traffic data sets.

Select $t_{max} = 300$. In Figure 14, the horizontal axis is the number of session packets and the vertical axis is the duration of the session. Among them, the data sample capacity enhanced from 13687 to 300899, which enhanced by 21.98 times.

B. NEURAL NETWORKS BASED ON LONG AND SHORT SESSIONS

CNN can get features from shorter fragments, and the position of the features in the data fragments is not highly correlated, that is to say, CNN is a very effective prediction method for stationary time series [28]. LSTM is a cyclic neural network, which is more suitable for dealing with and predicting events with relatively long intervals and delays in time series, namely autocorrelation time series [29]. Therefore, in view of the conclusion above: the stability of shorter sessions and the self-correlation of longer sessions, a neural network is constructed based on the CNN layer and the LSTM layer, respectively, to classify normal traffic and abnormal traffic.

According to the optimal time window range, the sample set is classified with 100s as the window size. Because $t_{max} = 300$, three neural networks are trained. For different sample capacity, the larger one is sampled randomly to make it equal to the capacity value of the smaller one, k-fold validation was used to reduce the verification variance in all three groups

TABLE 4. Structure of model_2 & model_3.

Layer	Type	Filters/neurons	Stride	Activation Function	Other attributes
1	Conv_1D	4	6	sigmoid	—
2	Max_Pooling	—	4	—	—
3	Conv_1D	8	6	sigmoid	—
4	Max_Pooling	—	4	—	—
5	Conv_1D	8	6	sigmoid	—
6	Max_Pooling	—	4	—	—
7	Conv_1D	12	6	sigmoid	—
8	Max_Pooling	—	4	—	—
9	LSTM	6	—	tanh	dropout (0.3)
10	LSTM	8	—	tanh	dropout (0.3)
11	Dense	128	—	sigmoid	L2 (0.01) regularization
12	Dense	256	—	sigmoid	L2 (0.01) regularization
13	Dense	64	—	sigmoid	—
14	Dense	1	—	sigmoid	—

of experiments. The experimental CPU is i7-6700 and the GPU is NVIDIA Tesla K80. The specific parameters are shown in Table 2, the structure of model_1, model_2 and model_3 are shown in tables 3 and 4, the experimental results are shown in Fig. 15.

Figure 15 shows that in the first group of experiments, the samples are sessions with shorter duration. The PIMDL provides less information, cause the discrimination between normal traffic and abnormal traffic is less, which makes the experiment presenting over-fitting after 40 rounds, validation loss value begins to increase and accuracy begins to decline. Therefore, for model_1, the model at the 40th round is selected as the final model. In the second and third groups of experiments, because PIMDL provided abundant information, the discrimination between abnormal flow and normal flow was relatively high. When the experiment was carried out to 300 rounds, validation loss showed a convergence trend, and the neural network maintained a high accuracy for the classification of normal and abnormal flow. Therefore, for model_2 and model_3, the final model is selected as the final model. The accuracy of the three models in the test set is 86.347%, 91.104%, and 96.262%, respectively.

C. SIMULATION EXPERIMENT BASED ON IMPROVED INTRUSION DETECTION

1) IMPROVED INTRUSION DETECTION ALGORITHM

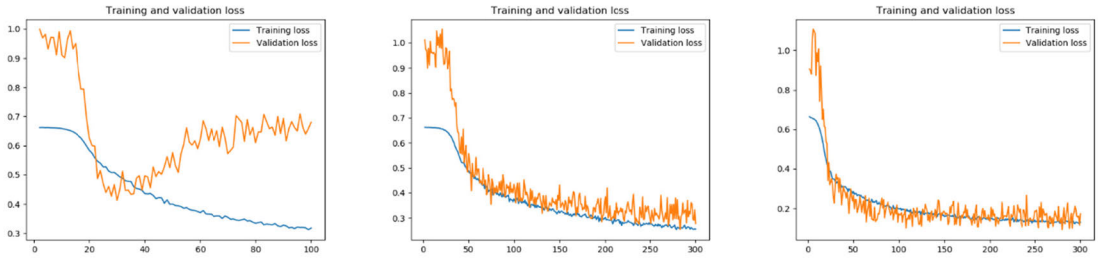
Segmented data enhancement of abnormal traffic makes the neural network learn a lot of abnormal characteristics of segments, so that three groups of neural networks can fit the characteristics of PIMDL duration in (0,100), (100,200), (200,300), respectively. However, the following problems need to be solved when it is used in intrusion detection:

It is necessary to predict the duration of the session in order to get its complete information. In real traffic generation, the duration of the session cannot be predicted in advance, which makes it possible to use the neural network to classify the traffic only when the session is over, which greatly reduces the real-time of intrusion detection;

In the training process of the neural network, the abnormal flow is enhanced by piecewise data. The three groups of neural networks obtained only fit the piecewise characteristics of PIMDL, not the overall characteristics. Its application needs further treatment.

Aiming at the above problems, a collaboration mechanism between neural networks is designed to evaluate the network traffic synthetically to determine the abnormal traffic. That is to say, session PIMDL is graded sequentially by three groups of neural networks (score is between 0 and 1, calculated by sigmoid activation function of the last layer). According to the three groups of model scores of session PIMDL, sessions are mapped to the three-dimensional coordinate system, and traffic is classified by pre-trained an SVM [30] model. The specific flow chart is shown in Figure 16 (For clearer typesetting, Fig 16 is on the next page).

The algorithm flow shown in Figure 16 is divided into two modules: session scoring module and abnormal detection module. The main task of the session scoring module is to use model_1, model_2, and model_3 for scoring when the session exceeds 100s, 200s and 300s. If the hopping interval of session data packets is too long (e.g. the data packets do not appear in 100-200 s), the lower model (e.g. model_1)



(a) epoch-acc (numbers from left to right are 1, 2 , 3, respectively.)

(b) epoch-loss (numbers from left to right are 1, 2 , 3, respectively.)

FIGURE 15. In three groups of experiments, the trend of loss and accuracy of neural network with epoch increasing.

should be re-used for further prediction to ensure that all sessions are scored in the order of model_1, model_2, and model_3. When the session exceeds 300 seconds, all arrival time records of the session are cleared to keep the session duration in the (0s, 300s) interval. For long sessions, one of the models may be scored more than once because of repeated emptying of arrival times. The main task of abnormal detection module is to obtain all the sessions with three model scores at the end of the detection period, map three scores (if the session scored more than once, then take its average as the final score) to the three-dimensional coordinate system, and classify them using the pre-trained SVM model to achieve the purpose of intrusion detection. Pre-trained SVM model acquisition method is: the above training data (60% of the initial training traffic is injected into the normal traffic) is trained by improved intrusion detection algorithm, and the mapping of normal and abnormal speech annotated in the three-dimensional coordinate system is obtained periodically. The mapping of normal and abnormal sessions for 10 min, 20 min, 30 min and 40 min in the three-dimensional coordinate system is shown in Figure 17.

In Figure 17, the X, Y and Z axes represent the scores of the three models, respectively. As can be seen from Figure 17, abnormal sessions and normal sessions are kept at a certain distance, and SVM can get an ideal classification result. The SVM model is trained with two kinds of data at the 2400s. The number of normal and abnormal session samples were 15740 and 216, respectively. Samples are divided into training and test sets according to the ratio of 7:3, and Gauss kernel function is selected as its kernel function, gamma = 10. The accuracy of the final model in training and test sets is 98.69% and 98.65%, respectively.

V. EXPERIMENTAL RESULTS ANALYSIS

Based on the flow chart of the improved intrusion detection algorithm in Figure 17, a simulation experiment is designed to verify its detection index. CICIDS-2017 and CTU-13 were injected into MAWI network traffic. In order to keep a gap with training data, the abnormal traffic was measured by 40% of the initial traffic, and the normal traffic was selected by MAWI from 00:00 to 02:00 on April 9. According to the algorithm flow, the accuracy (ACC), detection rate (DR) and false alarm rate (FAR) are counted in each detection period. The three definitions are as follows:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$DR = \frac{TP}{TP + FN} \tag{9}$$

$$FAR = \frac{FP}{FP + TN} \tag{10}$$

Among them, TP denotes the number of abnormal sessions correctly identified, TN denotes the number of normal sessions correctly identified, FP denotes the number of normal calls incorrectly identified, and FN denotes the number of abnormal sessions incorrectly identified.

The running time of the system is 3500s and the detection period is 100s, and the statistics start from 500s. Each detection period counts the above three indicators. The results are shown in Figure 18, it shows the changes of ACC, DR and FAR with running time, their mean values are 98.90%, 90.13%, and 0.96%, respectively. At the end of the system, the number of normal sessions and abnormal sessions is 16 442 and 405, respectively, and the abnormal traffic accounts for 2.4% of the total traffic. This shows that the improved

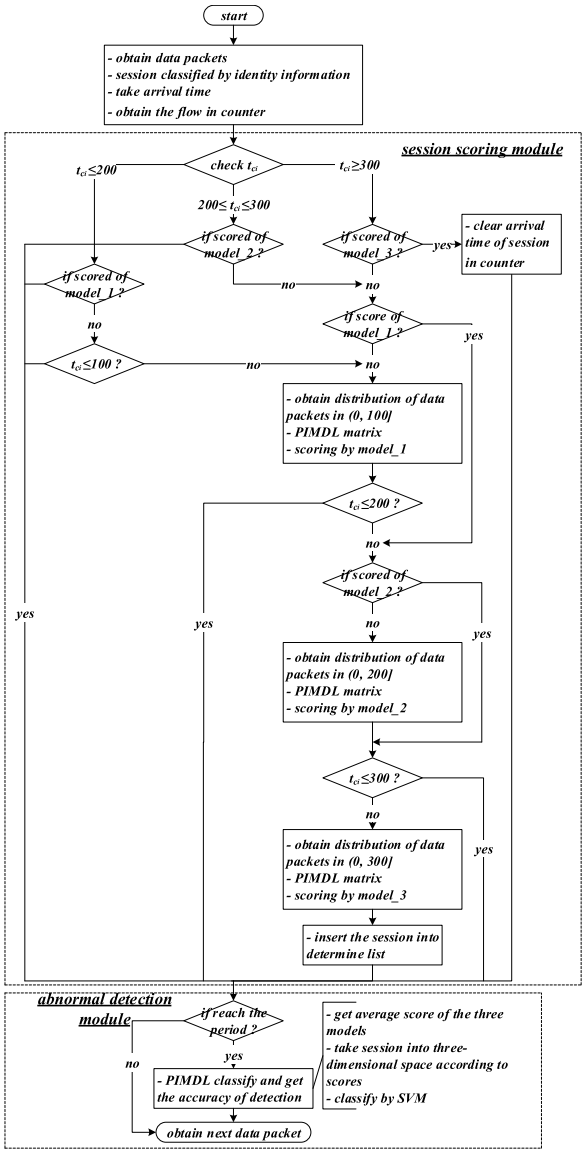


FIGURE 16. Intrusion detection specific process.

algorithm can still have higher accuracy and detection rate and lower false alarm rate when the abnormal session size is much smaller than the normal session size.

In order to test the improvement of the proposed algorithm in computing speed, NSL-KDD-type datasets are attempted to build from initial traffic, using MAWI data set (from 00:00 to 02:00 on April 9) as experimental data. As benchmark algorithm, all 41-dimensional data (e.g. first two seconds traffic information, first 100 connection traffic information) are obtained in the process of real traffic generation. When an attribute cannot be acquired due to incomplete information, it is skipped to acquire the attribute. The computational time required to acquire attributes in the process of generating simulated traffic is counted. The experimental results are shown in Figure 19.

Figure 19 shows the calculation time of the improved algorithm. The horizontal axis represents the flow generation time, and the vertical axis represents the solution time of the algorithm with a time interval of 0.1s before the last one. The average processing time of the benchmark algorithm and the improved algorithm is 1.6276s and 0.8899s per unit under 900s, and the standard deviation is 0.4720 and 0.1595, respectively. The experimental results show that compared with the benchmark algorithm, it needs to maintain traffic information lasting for two seconds, including 100 connections, and frequently performs queries and calculations, which results in a large amount of computational burden. The improved algorithm only obtains identity information and arrival time, so it has less computational burden and volatility. In addition, some attributes are skipped in the experiment, so the benchmark algorithm will face a more serious computational burden in the real situation.

Because previous algorithms rely too much on high-level protocol data, they cannot accurately identify attack traffic such as camouflage protocol. In order to verify this limitation, the random shuffle method is used to replace the attributes of the camouflaged samples with the attributes of other samples to simulate the camouflage behavior. Record ACC once for every camouflage attribute. The proportion of camouflaged traffic was 20%, 40%, 60%, and 80%, respectively. NSL-KDD is selected as the data set, and the camouflage attributes to select the high-level protocol attributes of NSL-KDD, in the feature set provided by NSL-KDD, there are two types of attributes. The first type is high-level attributes, which are easy camouflaged, such as login times, access times. The second type is the attributes of the data link layer, which relies on the traffic information of other users in the data link layer and is not easy camouflaged, such as the traffic information of the first two seconds, the information of the first 100 connections, the order of camouflage attributes were numbered, as shown in Table 5, the method of [8-11] is used as the benchmark algorithm. The experimental results are shown in Figure 20 and Table 6.

Because these high-level protocols are not used for training, the accuracy of the improved algorithm is not affected by the camouflage protocol, which is stable at 98.90%. Because this experiment only needs to prove the problem of accuracy decline, there is no evaluation of DR and FPR indicators.

In order to further explore the accuracy improvement effect of the proposed improved algorithm in small sample abnormal traffic classification task, reduce the scale of abnormal traffic and detect its accuracy, the above benchmark algorithm is also used as a reference, and ROC curve and AUC are used to evaluate its detection accuracy. The ROC curve is the representation of TPR (True Positive Rate) and FPR (False Positive Rate) under different thresholds. They are valued in the same way as DR and FAR.

AUC (Area under Curve) is the area under the ROC curve, which is between 0 and 1. When a positive sample and a negative sample are randomly selected, the probability that the current classification algorithm ranks the positive sample

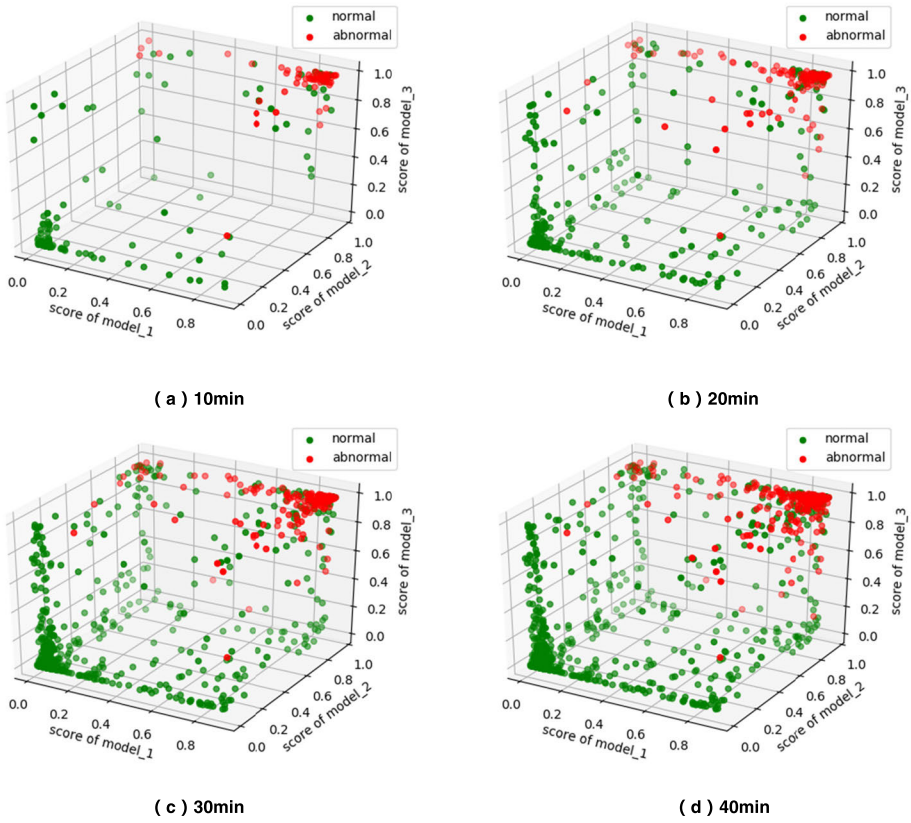


FIGURE 17. Mapping of normal & abnormal PIMDL in scoring space of three-dimensional model.

TABLE 5. High level protocol in NSL-KDD data set.

num	attribute	num	attribute	num	attribute	num	attribute
1	protocol_type	6	logged_in	11	land	16	num_file_creations
2	service	7	num_compromised	12	wrong_fragment	17	num_shells
3	flag	8	root_shell	13	urgent	18	num_access_files
4	src_bytes	9	su_attempted	14	hot	19	num_outbound_cmds
5	dst_bytes	10	num_root	15	num_failed_logins	20	is_guest_login

TABLE 6. Accuracy under camouflage (average / minimum accuracy).

camouflage ratio	20%	40%	60%	80%
T-S[8]	95.02% /93.98%	90.36% /88.37%	85.48% /82.43%	80.70% /76.60%
CNN[9]	94.60% /94.40%	93.55% /93.10%	92.88% /92.29%	92.13% /91.35%
STL[10]	96.21% /95.50%	93.38% /91.89%	90.31% /87.97%	87.19% /84.18%
SHIA[11]	90.48% /90.33%	90.26% /89.98%	90.11% /89.73%	89.87% /89.27%
Improved	98.90%			

ahead of the negative sample according to the calculated Score value is the AUC value. As a numerical value, AUC can evaluate the quality of classifier intuitively.

In the experiment, abnormal traffic in the NSL-KDD data set is sampled randomly, and its scale is reduced to 2.40% in the above experiment. The accuracy of the traditional and

TABLE 7. AUC with different experimental numbers.

Number	1	2	3	4	5	6	7	8	9	0	mean
T-S[8]	0.9794	0.9795	0.9794	0.9781	0.9766	0.9703	0.9794	0.9789	0.9787	0.9796	0.9780
RNN[9]	0.9683	0.9680	0.9687	0.9660	0.9644	0.9698	0.9665	0.9695	0.9669	0.9649	0.9673
STL[10]	0.9495	0.9476	0.9482	0.9525	0.9511	0.9506	0.9509	0.9493	0.9476	0.9504	0.9498
SHIA[11]	0.9346	0.9344	0.9463	0.9584	0.9466	0.9526	0.9609	0.9343	0.9412	0.9198	0.9429
Improved						0.9835					

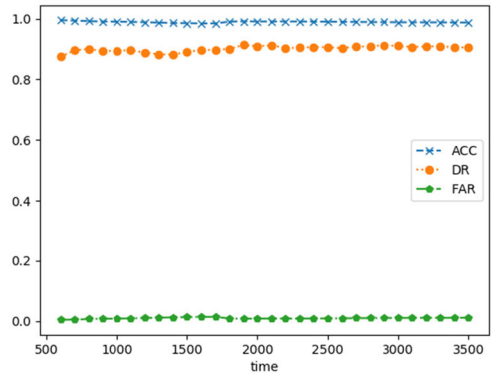


FIGURE 18. Results of simulation experiments.

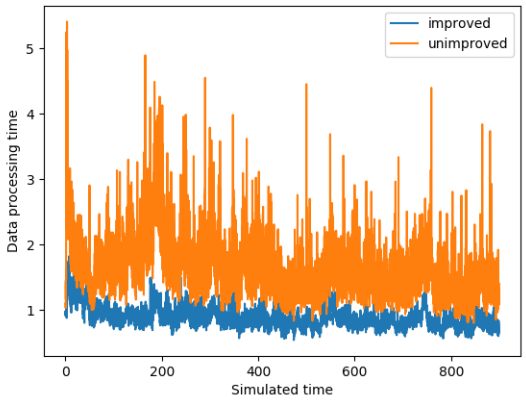


FIGURE 19. Comparing the processing time of the improved algorithm with that of the traditional algorithm as the simulation time goes on.

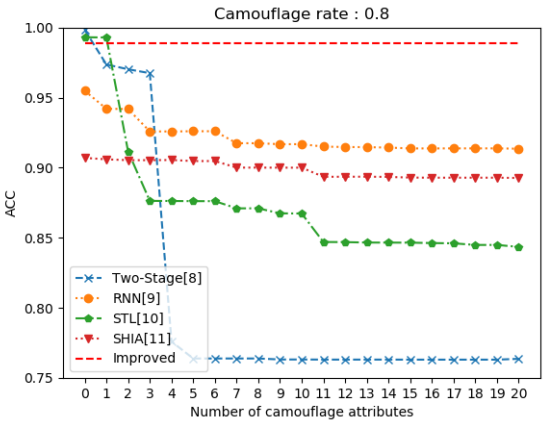


FIGURE 20. Accuracy changes with the increase of camouflage attributes (the camouflage rate was 80%).

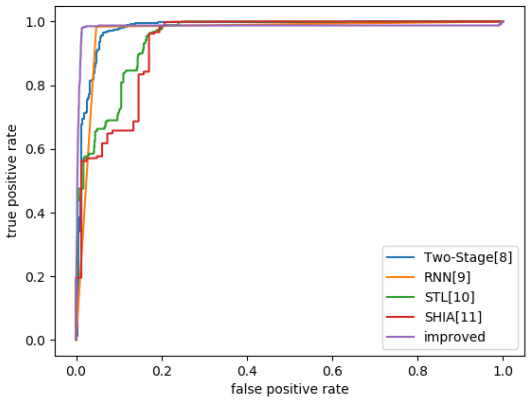


FIGURE 21. ROC comparison in the small sample (other experimental results are similar: the performance of ROC curve of traditional algorithm is weaker than that of improved algorithm).

improved algorithms for small sample abnormal traffic classification is explored. The experiment was conducted 10 times to ensure that random sampling does not introduce bias. The experimental results are shown in Figure 21, the AUC performance of each numbered experiment is shown in Table 7.

Because the ideal target TPR = 1, FPR = 0, the closer the ROC curve is to (0,1) points and the more it deviates from the 45 degrees diagonal, the better the effect will be. The experimental results show that the performance of AUC and ROC curves of the traditional algorithm is lower than that of the improved algorithm in small sample abnormal traffic detection tasks. This is because when training small samples, the small capacity of the neural network can not generalize all the features of this class. When classifying unlabeled data, the features may not be learned by the neural network, resulting in larger errors.

VI. CONCLUSION

Aiming at the difference of interaction modes between traffic ports in the complex network environment, a PIMDL model is proposed to quantify the mapping of interaction modes between ports at the link layer. On the basis of verificating

the feasibility of PIMDL, and neural networks construction method based on CNN and LSTM are designed to recognize normal and abnormal PIMDL, and intrusion detection system is carried out through multi-model evaluation mechanism. Compared with the traditional feature sets that previous studies depend on, this paper trains by acquiring the arrival time of traffic data packets, improves the computational efficiency, detection rate in the case of camouflage information, the accuracy of anomaly detection in small samples. However, this method has high spatial complexity because it maintains a time series for each session. Next, the mapping model of sessions in different autonomous domains at the data link layer will be established by bifurcation and chaos theory, and the effect of actual distance on PIMDL will be explored to classify abnormal traffic more accurately.

REFERENCES

[1] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.
[2] F. Zhang and D. Wang, "An effective feature selection approach for network intrusion detection," in *Proc. IEEE 8th Int. Conf. Netw. Archit. Storage*, Xi'an, China, Jul. 2013, pp. 307–311.
[3] S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, "The UCI KDD archive of large data sets for data mining research and experimentation," *ACM SIGKDD Explor. Newslett.*, vol. 2, no. 2, pp. 81–85, Dec. 2000.
[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6.
[5] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. MilCIS*, Canberra, ACT, Australia, Nov. 2015, pp. 1–6.
[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, Portugal, Portuguese, Jan. 2018, pp. 108–116.
[7] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012.
[8] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.
[9] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[10] A. Y. Javaid, Q. Niyaz, and W. Q. Sun, "A deep learning approach for network intrusion detection System," presented at the BICT, 2015.
[11] 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.
[12] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.
[13] W. Wang, M. Zhu, X. Zeng, X. Z. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. ICOIN*, Da Nang, Vietnam, Jan. 2017, pp. 712–717.
[14] F. Jiang, Y. S. Fu, B. B. Gupta, and F. Lou, "Deep learning based multi-channel intelligent attack detection for data security," *IEEE Trans. Sustain. Comput.*, to be published, doi: 10.1109/TSUSC.2018.2793284.
[15] B. Kolosnjaji, G. Eraisha, G. Webster, A. Zarras, and C. Eckert, "Empowering convolutional networks for malware classification and analysis," in *Proc. IJCNN*, Anchorage, AK, USA, May 2017, pp. 3838–3845.
[16] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Comput. Secur.*, vol. 29, no. 1, pp. 124–140, 2010.
[17] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.
[18] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proc. ISI*, Beijing, China, Jul. 2017, pp. 43–48.
[19] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick* (Lecture Notes in Mathematics). Berlin, Germany: Springer Verlag, 1981, pp. 366–381.
[20] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A, Gen. Phys.*, vol. 33, no. 2, pp. 1134–1140, Feb. 1986.
[21] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," presented at the Project Usenix Freenix Track, San Diego, CA, USA, Jun. 2000.
[22] F. Takens, "On the numerical determination of the dimension of an attractor," in *Dynamical Systems and Bifurcations* (Lecture Notes in Mathematics). Berlin, Germany: Springer Verlag, 1985, pp. 99–106.
[23] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Phys. D, Nonlinear Phenomena*, vol. 9, no. 1, pp. 189–208, Oct. 1983.
[24] S. Garcia, M. Grill, H. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014.
[25] J. Benesty, J. Chen, and Y. Huang, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*, vol. 2. Springer, 2009, pp. 37–40, doi: 10.1007/978-3-642-00296-0_5.
[26] M. K. Clayton and B. D. Hudelson, "Confidence intervals for autocorrelations based on cyclic samples," *J. Amer. Stat. Assoc.*, vol. 90, no. 430, pp. 753–757, 1993.
[27] A. Liu and B. Sun, "The improved model for anomaly detection based on clustering and dividing of flow," presented at the IEEE DSC, Hangzhou, China, 2019.
[28] S. A. Amirshahi and X. Y. Stella, "CNN feature similarity: Paintings are more self-similar at all levels," presented at the CVCS, 2018.
[29] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[30] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifier," presented at the Workshop Comput. Learn. Theory. 1992.

AO LIU was born in Handan, Hebei, China, in 1996. He received the B.S. degree in computer science and technology from the Hebei University of Engineering, Handan, in 2017. He is currently pursuing the M.E. degree in computer science and technology with the Tianjin University of Technology, Tianjin, China. His research interests include cyberspace security, network traffic intrusion detection, and machine learning.

BIN SUN was born in Tianjin, China, in 1981. He received the M.S. degree in computer science and technology from the Tianjin University of Technology, Tianjin, in 2010, and the Ph.D. degree in operational research and cybernetics from Nankai University, Tianjin, in 2013. He is currently a Lecturer with the School of Computer Science and Engineering, Tianjin University of Technology. His research interests include collaborative optimization, transportation decision-making, and cyberspace security.