IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 18, 2019, accepted October 7, 2019, date of publication October 11, 2019, date of current version October 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946708

Analysis of Multi-Types of Flow Features Based on Hybrid Neural Network for Improving Network Anomaly Detection

CHENCHENG MA, XUEHUI DU, AND LIFENG CAO
National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450000, China
Zhengzhou Science and Technology Institute, Zhengzhou 450000, China

Corresponding author: Xuehui Du (dxh37139@sina.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803603 and Grant 2016YFB0501901, and in part by the National Natural Science Foundation of China under Grant 61502531, Grant 61702550, and Grant 61802436.

ABSTRACT Security issues of large-scale local area network are becoming more prominent and the anomaly detection for the network traffic is the key means to solve this problem. On the other hand, it is a challenge to extract effective and accurate traffic features for anomaly detection. In order to resolve this challenge, multi-types of network flow features are designed and analyzed in the present study. These features include sequence packet features, general statistical features and environmental features, which can profile the characteristics of network flows accurately. Moreover, a method based on the hybrid neural network is proposed to detect anomaly by analyzing these features. One-dimensional convolutional network is implemented to analyze the sequence features in the hybrid neural network, while deep neural networks are utilized to learn the characteristics of high-dimension feature vectors including general statistical features and environmental features. The method can make comprehensive analysis for network anomaly detection. Two datasets of ISCX-IDS-2012 and CIC-IDS-2017 are carried out to evaluate the performance of the proposed method and other similar algorithms. The present study shows that the comprehensive performances of the proposed method are better than that for others algorithms. It is concluded that the proposed method can be applied for the anomaly detection applications with reasonable performance.

INDEX TERMS Anomaly detection, hybrid neural network, network flow feature.

I. INTRODUCTION

Access security of the Large-scale Local Area Network (LLAN) is currently a network security issue that needs urgent attention [1].With the advent and development of network systems, most military and government institutions have built large-scale local area networks to enhance the corresponding office convenience. Studies show that the LLAN is a widely adopted network organization mode. Meanwhile, important LLANs store a large amount of private and sensitive information so that they are frequently faced with malicious acts of malefactors [2]. Therefore, security issues of local area networks are of significant importance and they have become increasingly prominent.

Network anomaly detection is the main means of maintaining the network security [3]. Based on specific characteristics of the network traffic, a wide variety of anomaly detection methods and models has been developed. Different assumptions, including the sequential characteristics of the network traffic [4], statistical characteristics of the traffic [5] and the overall environmental distribution of the traffic [6], are made in this regard. However, almost all of these models only analyze the network traffic from a single characteristic. In fact, these methods only analyze one of the traffic characteristics among the sequence, statistics and the environmental properties of the network. On the other hand, the network traffic has different characteristics from different perspectives so that it is impossible to fully describe the characteristics of the network traffic [7].

Aiming at solving these challenges, the present study is focused on the design and analysis of features of multi-types.

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

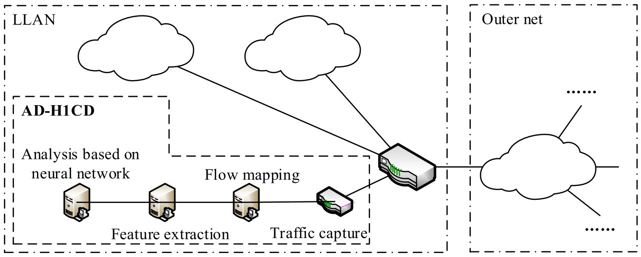


FIGURE 1. Deployment of the anomaly detection system.

One-dimensional Convolution Neural Network (1D-CNN) and the Deep Neural Network (DNN) are combined in these regards. It is intended to propose a new Anomaly Detection method for network flows based on the Hybrid neural network comprised of the 1D-CNN and the DNN (AD-H1CD). Fig.1 shows that the proposed method deploys the anomaly detection system at the access gateway of the LLAN to detect abnormal network flows during the communication between internal and external networks. The system contains 4 parts, including the traffic capture, flow mapping, feature extraction and the analysis part which is based on the neural network. AD-H1CD extracts sequence packet features of the single flow, general statistical features of the single flow and environmental features of flows, which represents the traffic characteristics of the sequence, statistics and the overall environmental distribution, respectively. The hybrid network based on 1D-CNN and DNN can analyze these characteristics. In summary, the main contributions of the present study are as follows:

(1) The three dimensions of sequential, statistical and environmental characteristics of the network flow are innovatively proposed, which provides a more comprehensive and detailed portrayal of the network flow. Moreover, three feature sets are designed, which contain some new features and extraction methods.

(2) A new hybrid neural network comprised of the 1D-CNN and DNN schemes is presented, which can effectively analyze three dimensions of sequential, statistical and environmental characteristics of network flows for the anomaly detection.

(3) A scheme of the flow-sliding window is put forward to improve the extraction of environmental features of the flow.

(4) The performance of the proposed method is evaluated on two datasets in order to show its effectiveness and superiority.

The present study is organized as the following: Section 2 is dedicated to review related works of the traffic anomaly detection, while analyzing the model construction of the hybrid neural network and the sub-parts are presented in section 3. Moreover, the architecture of the proposed AD-H1CD model and the formulation of multi-types network flows features are described in section 4. Section 5 contains the experimental part, including the parameter selection for the AD-H1CD model, verifying the proposed model and comparison with similar methods based on two published datasets (ISCX-IDS-2012 and CIC-IDS-2017).

II. RELATED WORK

The existing popular anomaly detection models can be classified into three types, including the general probability statistical model, general machine learning model and the neural network model. Compared to general machine learning models, the neural network model can achieve the deep learning. In other words, the neural network model can capture much more deep characteristics of the traffic. Although the neural network is regarded as one of machine learning methods, the neural network model is actually different from the conventional machine learning models that can only learn shallow features [3]. It should be indicated that the model selection usually determines the feature type.

The general probability statistical model is usually applied to analyze features of the environmental distribution of the network traffic. This model is applied in the feature of traffic matrix [8] and the feature of multi-flows [9]. Furthermore, the distribution feature of packets is the most used feature of the general probability statistical model. Researches in this area include the decomposition and analysis of the anomaly matrix based on the Principal Component Analysis (PCA) [10], anomaly analysis based on the probability hypothesis, histogram-based analysis [11], flooding attack detection based on the feature entropy [12] and the anomaly analysis based on the counting statistical information of packets [13]. Moreover, some researches applied the general probability statistical model to analyze the sequence characteristics of the traffic [14]. However, this model is rarely used for such applications. Studies showed that entropy-based methods have limited effectiveness on the general probability statistical model so that employing only the entropy distribution feature cannot always be useful for the appropriate anomaly detection [15]. Ringberg *et al.* [16] analyzed the inevitable difficulty of the PCA in the anomaly detection and expressed the complexity of confirming the dimensions of the normal subspace.

The general machine learning method is usually applied to analyze the spatial or temporal feature of packets, such as bytes or time series. Several methods have been proposed so far based on the general machine learning method. More specifically, anomaly detection methods for packet sequences based on the Markov chain [17], cluster [18], decision tree [19], Bayesian Markov chain [20] and based on the Support Vector Machine (SVM) [21] can be mentioned in this regards. Moreover, researches developed different detection schemes based on spatio-temporal flow features, including the anomaly detection based on K-Nearest Neighbor (KNN) [22] and the anomaly detection based on the SVM [23], to investigate the sequential characteristics of the traffic.

Since the neural network can learn high-dimension features, many researchers have used it to detect network anomalies and achieved reasonable results. For this type of model, general statistical features of the single flow are often used. Radford *et al.* [24] utilized Netflow data to analyze the flow feature vector based on the recurrent neural network. Hguyen *et al.* [25] investigated differences between the output vector from the auto-encoder neural network and the initial flow vector to find out the anomaly. Moore and Vann [26] implemented the 2D-CNN to learn hardware data in network facilities and detect possible anomaly. Other methods such as the auto-encoder neural network [27], Long and Short Time Memory network (LSTM) [28] and the CNN [29] are also used to detect the anomaly through encoding the traffic. Furthermore, some researches have been carried out based on the environmental feature of multi-flows [30], which applied the DNN to learn the characteristics of the network flow.

Recently, application of the hybrid neural network for the anomaly detection has attracted many researchers. Wang *et al.* [31] converted flow packets into a picture and learned the characteristic of packet bytes based on the 2D-CNN, while learning the characteristic of the packet sequence based on the LSTM so that they simultaneously learned two characteristics of spacing and timing. Zeng *et al.* [32] applied the hybrid neural network comprised of the 1D-CNN, LSTM and the Stacked Auto-Encoder (SAE) to study traffic features and select the best feature vectors as the label result. However, these hybrid neural networks only analyze the characteristics of packet bytes sequence so that advantages of the hybrid structures are not employed appropriately. The key to modeling traffic anomaly detection is to select the model that adapts the traffic characteristics and choose the appropriate traffic feature set. Unlike conventional methods, the proposed method makes the comprehensive analysis for the network flow to perform the anomaly detection more accurately. Compared with the existing methods that only capture the single-type features, the proposed hybrid neural network can study multi-type characteristics of the sequence, statistics and the environment. It is expected that application of the hybrid neural network can result in more reasonable and more effective detections.

III. NETWORK FLOW ANOMALY DETECTION METHOD

The proposed detection method for the network flow anomaly combines the analysis of sequential, statistical and environmental characteristics, which can remarkably improve the anomaly detection. This section introduces the proposed method and analyzes it from different aspects, including the overall framework, flow mapping, various feature extractions and neural network construction. These feature designs are conducted based on some common features. The specific feature set will be described in the next section.

A. FRAMEWORK

1D-CNN and DNN methods are combined in the present study to propose the AD-H1CD method, which is a network flow anomaly detection method based on the hybrid neural network. The AD-H1CD algorithm maps the data packets to the network flow, updates sequence packet features of each flow in real time and periodically calculates the environmental features of flows. When a flow is ended, the AD-H1CD calculates general statistical features of the flow and generates three types of features to the hybrid network to analyze whether the flow is an abnormal flow or not. Fig.2 illustrates that the AD-H1CD scheme is mainly divided into 5 modules as the following:

(1) Flow mapping and processing: the main function of this module is to form a flow ID for the traffic packet, map it to the corresponding network flow and periodically detect whether there is a timeout flow or not.

(2) Sequence Packets Features (SPF) of the single flow processing: this module extracts m-dimension packet features of first n packets of the network flow in real time and forms the corresponding tensor with the appropriate dimension (n, m).

(3) General Statistical Features (GSF) of the single flow processing: a general feature vector of the flow is obtained in this module through performing a statistical analysis.

(4) Environmental Features (ENF) of the flow processing: this module periodically calculates environment of the flow distribution based on information of active flows in the flow-sliding window and generates the environmental features of ended flows.

(5) Hybrid neural network construction: This module is proposed to analyze flows, where 1D-CNN and DNN are applied for the analysis of multi-type features of network flows to determine the flow anomaly.

B. FLOWS MAPPING AND PROCESSING

The network flow mapping mechanism extracts flow elements for each packet and forms a flow ID, which it uniquely identifies a certain flow and it is used to map packets to corresponding flows. Fig.3 shows schematic network flow mapping, where the solid line indicates the time length that the flow actually experiences, while the packet at the solid arrow indicates that the last packet is mapped to the flow. Moreover, the dotted line shows the time length that the flow may still run. The length is limited and the longest length is the timeout period set in the timeout detection. It should be indicated that in the present study, the timeout period is set to 60 seconds. Therefore, when the dashed arrow does not reach the current moment on the time axis, it is inferred that the stream has timed out.

Network flows mapped by this method are mainly divided into TCP/UDP flows and ICMP flows. These three protocol network flows can basically cover the anomaly situations that occur in large LANs. Specific definitions of the network flow are as follows:

Definition 1 (Network Flow): A bidirectional sequence of packets from a pair of IP entities interacting on the same IP layer protocol over the same network in a certain period of time, which mainly consists of TCP/UDP and ICMP flows. Furthermore, the flow ID is applied to identify a certain flow.

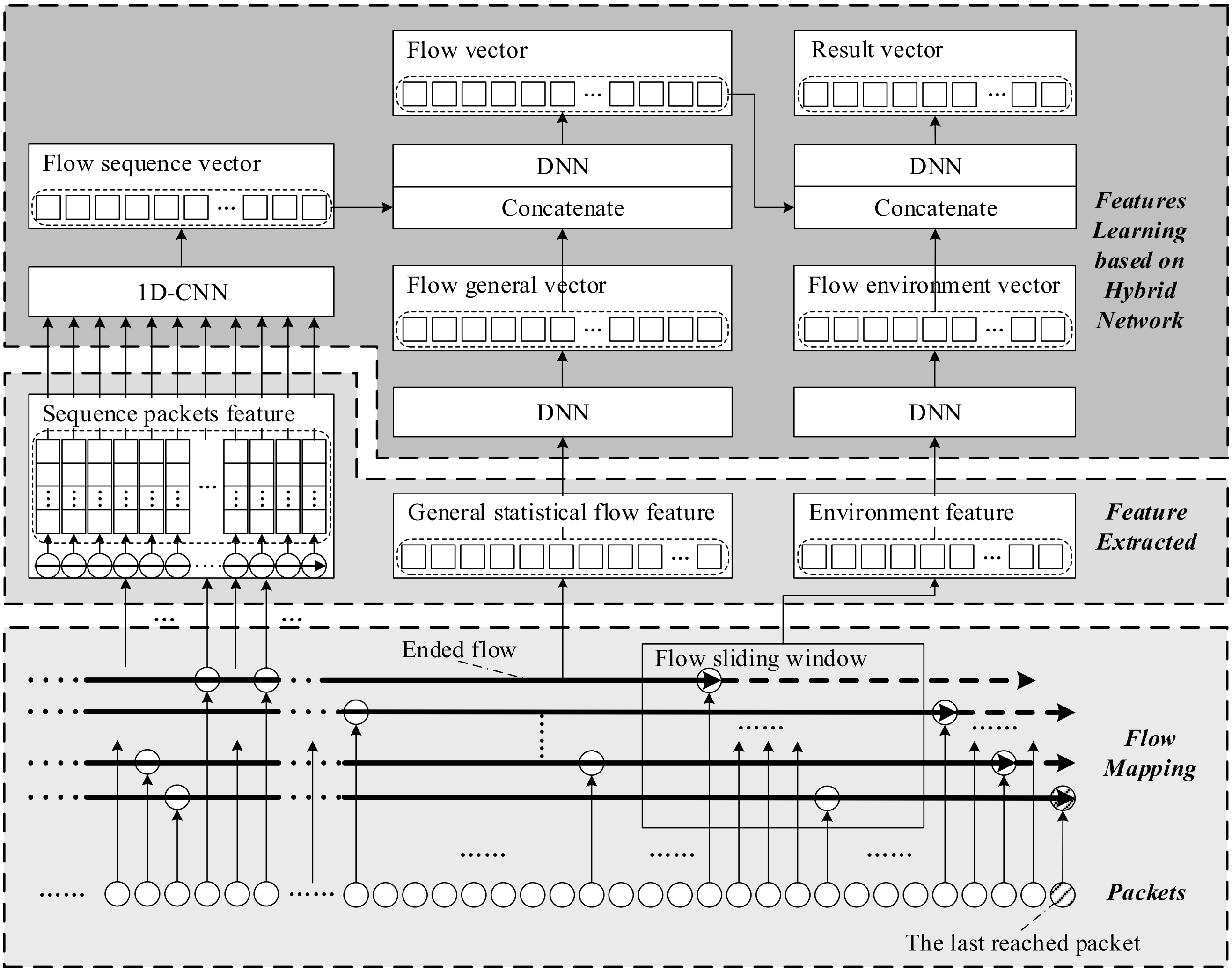


FIGURE 2. Schematic framework of the AD-H1CD.

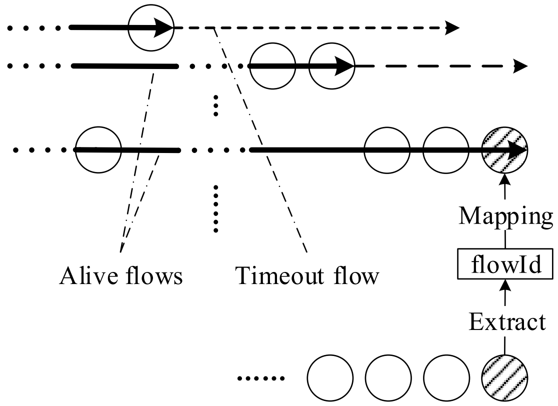


FIGURE 3. Network flows mapping.

Definition 2 (TCP/UDP Network Flow): The network flows with transport layer protocol of TCP/UDP, and the flow is identified by a quintuple, flowId<sIP, dIP, sPort, dPort, TCP/UDP>.

Definition 3 (ICMP Network Flow): Network flows with transport ICMP layer protocol and the flow is identified by the triplet, flowId<sIP, dIP, ICMP>.

Algorithm 1 shows that the network flow mapping and processing are mainly divided into three parallel sub-algorithms and two shared pools.

The first loop is used to capture network packets in real time, extract flow IDs for new packets and map them to the corresponding flows. If the flow is a new flow, it is added to the first shared pool entitled by the *aliveFlows*, where *aliveFlows* is employed to store network flows that are still alive. Then the Sequence Packet Feature (SPF) of the flow is updated.

The second loop is used to periodically calculate Environmental Feature (ENF) of flows based on flow sliding window information through *aliveFlows* and is added to the second shared pool *ENFs*.

The third loop is used to detect whether the network flow has ended or not. For the terminated network flow,

Algorithm 1 Flow Mapping and Processing

Output: predicting result of ended flows

```
aliveFlows ← []
ENFs ← []
while(pac, ts←capturePackets()): // capture packets of TCP, UDP and ICMP
    flow←flowMap(getFlowId(pac), ts); //flow mapping
    if flow is a new flow:
        aliveFlows.add(flow)
    end if
    flow←SPFUpdate(pac)
End while
while():// parallel processing for ENF calculation
    ENF←getSWValue(aliveFlows, ts)
    ENFs.add(ENF)
End while
while(): // parallel processing for ended flows
    for flow in aliveFlows:
        if flow is finished:
            SPF←flow.SPF
            GSF←calculateGSF(flow)
            ENF←getENF(ENFs,flow)
            prediction←NNModel(flow, SPF,GSF,ENF)
            output(prediction)
            aliveFlows.remove(flow)
        end if
    end for
End while
```

the algorithm calculates General Statistical Features (GSF) of the flow based on the SPF and searches the ENF corresponding to the flow from the shared pool *ENFs*. Finally, three types of eigenvalues of the ending flow are generated to the neural network model, and the prediction result of the flow is obtained.

C. EXTRACTION OF SEQUENCE PACKET FEATURE

A network flow has obvious characteristics of sequence and consists of successively arriving packets. Based on this premise, the present study intends to analyze network flows through sequence packet features of a single flow, to obtain laws of network flows.

Definition 4 (Sequence Packet Features (SPF) of the single flow): Extracting the 7-dimensional packet features from the first n packets of a network flow, and sequentially connecting packet feature vectors into a two-dimensional tensor, which is called the SPF. It should be indicated that the SPF of $Flow_i$ is recorded as SPF_i, while the m^{th} feature of the n^{th} sequence packet of $Flow_i$ is recorded as $SPF_i.S_m^{(n)}$.

Fig.4 indicates that when a packet arrives, the corresponding packet feature $(S_1, S_2, \ldots, S_m)$ is updated to the SPF of the flow. When the flow ends, the feature vectors formed by first n packets are sequentially connected to form the SPF.

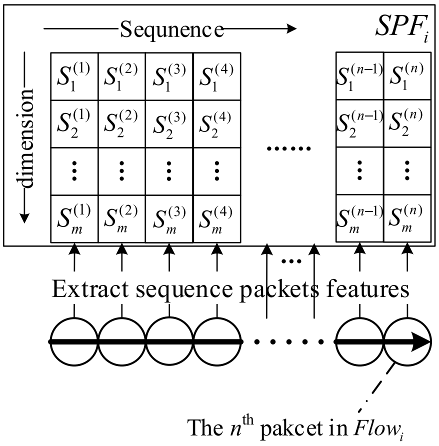

Sequnence
SPF_i
dimension
$S_1^{(1)}$ $S_1^{(2)}$ $S_1^{(3)}$ $S_1^{(4)}$ $S_1^{(n-1)}$ $S_1^{(n)}$
$S_2^{(1)}$ $S_2^{(2)}$ $S_2^{(3)}$ $S_2^{(4)}$ $S_2^{(n-1)}$ $S_2^{(n)}$
$S_m^{(1)}$ $S_m^{(2)}$ $S_m^{(3)}$ $S_m^{(4)}$ $S_m^{(n-1)}$ $S_m^{(n)}$
Extract sequence packets features
The n^{th} pakcet in $Flow_i$

FIGURE 4. Extraction processing of the SPF.

It should be indicated that the selection of n is of significant importance and the optimum value is discussed in section 5.

Table 1 shows that each vector in the SPF is a 7-dimension vector. The features of S1, S4 and S5 are presented in the present study, while features of the others are common features in prior studies [11], [14]. It should be noted that the packet direction represents the packet direction with respect to the forward direction of the flow. The direction of the first packet of the flow is used as the forward direction. By only analyzing the header field and statistical features, the content payloads of packets are prevented from being parsed, which bypasses the difficulty of the anomaly detection for the encrypted traffic.

D. EXTRACTION OF GENERAL STATISTICAL FEATURES

The AD-H1CD model extracts general statistical features of a single flow and further analyzes flows from the perspective of integrity and improves the portrayal of the single flow. Definition 5 is the definition of General Statistical Features of a single flow (GSF).

Definition 5 (General Statistical Features of single flow, GSF). GSF is the general statistical features of a single flow, which is calculated based on the first n packets of the flow, with the dimension of 38. The GSF of $Flow_i$ is recorded as GSF_i

The set of GSF mainly includes the transport layer protocol type, the total number of packets of the flow and some statistical features of first n packets of the flow from the perspectives of the whole, forward, and backward. Statistical features mainly include number of packets, packet ratio, packet velocity, byte velocity, and time interval of packets and the number of IP bytes in four dimensions of maximum, minimum, mean and standard deviation. Table 2 shows that functions of min(), max(), mean() and std() are to obtain the minimum, maximum, average and standard deviation value of the array, respectively. The features of G3-G8 are presented in the present study, while features of the others are common features in prior studies [33]. It should be noted that the

TABLE 1. **Sequence packets features of single flow (SPF).**

Num	Feature Name	Symbol and calculation formula
S1	Packet direction	$D[i] = 0$ if $pac[i]$ is forward packet else $D[i] = 1$
S2	IP packet bytes	$PByte[i] = \text{len}(pac[i].\text{IPPac})$,$pac[i]$.IPPacis the IP packet data of the i^{th} packet in the flow
S3	IP packet header bytes	$HByte[i] = \text{len}(pac[i].\text{IPPac.header})$, $pac[i]$.IPPac.header
S4	Inter-arrival time of packets	$IAT[i] = t(pac[i]) - t(pac[i-1])$, $t(pac[i])$ is the arrival time of the i^{th} packet in the flow
S5	Inter-arrival time of packets in the same direction	$IATSD[i] = t(pac[i]) - t(pac[i-j])$, $j = min(i-k), k \in \{k \mid D[k] == D[i]\}$
S6	DF flag	Value of DF flag in the header of IP packet
S7	MF flag	Value of MF flag in the header of IP packet

G2 feature is the total number of packets of the flow, while calculations of G3-G38 are based on first n packets of the flow.

E. EXTRACTION OF THE ENVIRONMENTAL FEATURE

Anomaly behavior is often composed of multi-flows associations and independent analysis of the single-flow is difficult to fully grasp traffic patterns of the network traffic. Therefore, the present study extracts environmental features of flows based on flow sliding window in order to improve the profile of network flows.

1) FLOW SLIDING WINDOW

The basic principle of the flow sliding window of the AD-H1CD is as follows. The right limit of the window is the current time. The AD-H1CD periodically calculates environmental features of flows (for example, every 0.1s), and a flow is linked to the corresponding window by the arrival time of its last packet, thereby corresponding environmental features are obtained. Fig.5 shows that the environmental feature ENF_i is calculated based on a sliding window centering on the arrival time of the last packet of $Flow_i$ and extending forward and backward by a value of α.

The calculation of environmental features based on the information of active flows in the flow sliding window is based on the assumption that active flows can represent the environment of flows distribution. Fig. 5 illustrates that active flows are flows whose packets exist in the sliding window. Moreover, the flow sliding window length parameter α is a tunable parameter that is explored in subsequent experiments.

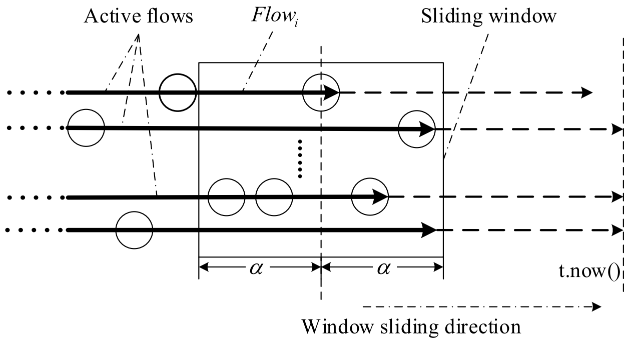


FIGURE 5. **Active flows in flow sliding window.**

2) DESIGN OF THE ENVIRONMENTAL FEATURE SET

The purpose of computing the environmental features is to analyze the current network flows distribution. On the one hand, it is helpful to analyze the emergence of flooding attacks. On the other hand, it can be combined with single-flow analysis to make more accurate detection for single-flow anomalies. The definition is as follows.

Definition 6 (Environmental Features of Flows, ENF): A 12-dimension environmental feature vector calculated from the distribution of active flows in the flow sliding window, which is corresponding to the flow based on statistical principles such as entropy. The ENF of $flow_i$ is recorded as ENF_i.

Table 3 shows the design of the environmental feature set. It is divided into three parts. The first part is the E1 feature, which describes the total number of active flows. The second part are the IP pair related features, including the number of IP pairs, maximum number of flows for an IP pair, ratio of the maximum number of flows for an IP pair and the entropy of the number of flows for IP pairs. The third part are features for the distribution of different transport layer protocol flows, including the number of different flows, ratio of the three types of protocol flows and the distributed entropy of the number of these three types flows. The function for E2 calculation in table 3, set(), is to find the largest subset of elements without repetition. Furthermore, E3-E5 and E12 features are proposed in the present study, while others are common features [6].

F. NEURAL NETWORK MODEL

The present study intends to construct a hybrid network based on 1D-CNN and DNN in order to learn the sequential, statistical and environmental characteristics of flows and the correlation between them to detect anomaly of the network flow.

1) HYBRID NEURAL NETWORK BASED ON 1D-CNN AND DNN

Based on dual characteristics of anomaly network flow with single-flow characteristics and environmental characteristics such as distributed attacks, the present study designs a hybrid neural network to meet the requirement of multi-types features analysis for flows anomaly detection. It analyzes

TABLE 2. General statistical features of single flow (GSF).

Num	Feature Name		Symbol and calculation formula
G1	Transport protocol		p
G2	Number of all the packets		$N_{allP}, N_{allP} \neq n$
G3-4	Number of packets in different direction for the first 10 packets	forward packets	$N_{FP}^{(10)} = \sum_{i=1}^{10} (1 \text{ if } D[i] == 0 \text{ else } 0)$
		backward packets	$N_{BP}^{(10)} = \sum_{i=1}^{10} (1 \text{ if } D[i] == 1 \text{ else } 0)$
G5-6	Number of packets in different direction for the first *n* packets	forward packets	$N_{FP} = \sum_{i=1}^{n} (1 \text{ if } D[i] == 1 \text{ else } 0)$
		backward packets	$N_{BP} = \sum_{i=1}^{n} (1 \text{ if } D[i] == 0 \text{ else } 0)$
G7-8	Ratio of the number of backward packets to forward packets for the first packets	first 10 packets	$R_P^{(10)} = \frac{N_{BP}^{(10)}}{N_{FP}^{(10)}}$
		first *n* packets	$R_P = \frac{N_{BP}}{N_{FP}}$
G9-12	Packet inter-arrival time	Minimum	$IAT\_min = \min(IAT[])$
		Maximum	$IAT\_max = \max(IAT[])$
		Mean	$IAT\_mean = \text{mean}(IAT[])$
		Standard deviation	$IAT\_std = \text{std}(IAT[])$
G13-16	IP packet bytes	Minimum	$PByte\_min = \min(PByte[])$
		Maximum	$PByte\_max = \max(PByte[])$
		Mean	$PByte\_mean = \text{mean}(PByte[])$
		Standard deviation	$PByte\_std = \text{std}(PByte[])$
G17-20	Forward packet inter-arrival time	Minimum	$FIAT\_min = \min(FIAT[]), FIAT[] = \{IATSD[i] \mid D[i] == 0\}$
		Maximum	$FIAT\_max = \max(FIAT[])$
		Mean	$FIAT\_mean = \text{mean}(FIAT[])$
		Standard deviation	$FIAT\_std = \text{std}(FIAT[])$
G21-24	IP packet bytes of forward packets	Minimum	$FPByte\_min = \min(FPByte[]), FPByte[] = \{PByte[i] \mid D[i] == 0\}$
		Maximum	$FPByte\_max = \max(FPByte[])$
		Mean	$FPByte\_mean = \text{mean}(FPByte[])$
		Standard deviation	$FPByte\_std = \text{std}(FPByte[])$
G25-38	Backward packet inter-arrival time	Minimum	$BIAT\_min = \min(BIAT[]), BIAT[] = \{IATSD[i] \mid D[i] == 1\}$
		Maximum	$BIAT\_max = \max(BIAT[])$
		Mean	$BIAT\_mean = \text{mean}(BIAT[])$
		Standard deviation	$BIAT\_std = \text{std}(BIAT[])$
G29-32	IP packet bytes of backward packets	Minimum	$BPByte\_min = \min(BPByte[]), BPByte[] = \{PByte[i] \mid D[i] == 1\}$
		Maximum	$BPByte\_max = \max(BPByte[])$
		Mean	$BPByte\_mean = \text{mean}(BPByte[])$
		Standard deviation	$BPByte\_std = \text{std}(BPByte[])$

TABLE 2. *(Continued.)* **General statistical features of single flow (GSF).**

G33-35	Number of IP packet bytes per second	all packets	$PByte\_psec = \frac{\sum_{i=1}^{n} PByte[i]}{\sum_{i=1}^{n} IAT[i]}$
		forward packets	$FPByte\_psec = \frac{\sum_{i=1}^{N_{FP}} FPByte[i]}{\sum_{i=1}^{N_{FP}} FIAT[i]}$
		backward packets	$BPByte\_psec = \frac{\sum_{i=1}^{N_{BP}} FPByte[i]}{\sum_{i=1}^{N_{BP}} FIAT[i]}$
G36-38	Number of packets per second	all packets	$Pac\_psec = \frac{n}{\sum_{i=1}^{n} IAT[i]}$
		forward packets	$FPac\_psec = \frac{N_{FP}}{\sum_{i=1}^{N_{FP}} FIAT[i]}$
		backward packets	$BPac\_psec = \frac{N_{BP}}{\sum_{i=1}^{N_{BP}} FIAT[i]}$

TABLE 3. **Environmental features of flows (ENF).**

Num	Feature Name		Symbol and calculation formula
E1	Number of active flows		$N_{AF} = \sum_{i=1}^{n}(1 \text{ if } Flow[i] \text{ is } Active\ Flow \text{ else } 0)$
E2	Number of active IP pairs		$N_{AIPs} = \text{len}(AIPs), AIPs = \text{set}(\{(AFlow[i].\text{IPs}) \mid 1 \le i \le N_{AF}\})$
E3	Maximum number of active flows for an IP pair		$n_{AIPs}\_max = max(n_{AIPs}), n_{AIPs}[j] = \sum_{i=0}^{N_{AF}}(1 \text{ if } AFlow[i].\text{IPs} == AIPs[j] \text{ else } 0)$
E4	Maximum ratio of number of active flows for an IP pair to the number of all active flows		$R_{AIPs}\_max = max(R_{AIPs}), R_{AIPs}[j] = \frac{n_{AIPs}[j]}{N_{AF}}$
E5	Information entropy of flow numbers distribution of active IP pairs		$H_{AIPs} = -\sum_{i=1}^{N_{AIPs}} R_{AIPs}[j] \cdot log\, R_{AIPs}[j]$
E6-8	Number of active flows with specific transport protocol	TCP	$N_{TCPF} = \sum_{i=1}^{N_{AF}}(1 \text{ if } Flow[i].\text{protocol} == \text{TCP else } 0)$
		UDP	$N_{UDPF} = \sum_{i=1}^{N_{AF}}(1 \text{ if } Flow[i].\text{protocol} == \text{UDP else } 0)$
		ICMP	$N_{ICMPF} = \sum_{i=1}^{N_{AF}}(1 \text{ if } Flow[i].\text{protocol} == \text{ICMP else } 0)$
E9-11	Ratio of number of active flows with specific transport protocol to number of all active flows	TCP	$R_{TCPF} = \frac{N_{TCPF}}{N_{AF}}$
		UDP	$R_{UDPF} = \frac{N_{UDPF}}{N_{AF}}$
		ICMP	$R_{ICMPF} = \frac{N_{ICMPF}}{N_{AF}}$
E12	Information entropy of active flow numbers distribution of IP pairs		$H_p = -R_{TCPF} \cdot log\, R_{TCPF} - R_{UDPF} \cdot log\, R_{UDPF} - R_{ICMPF} \cdot log\, R_{ICMPF}$

sequential characteristics and general statistical characteristics of the single network flow based on 1D-CNN and DNN. Moreover, it analyzes the environmental distribution characteristics of flows based on the DNN. Fig.6 shows that the hybrid neural network proposed in the present study consists of five parts:

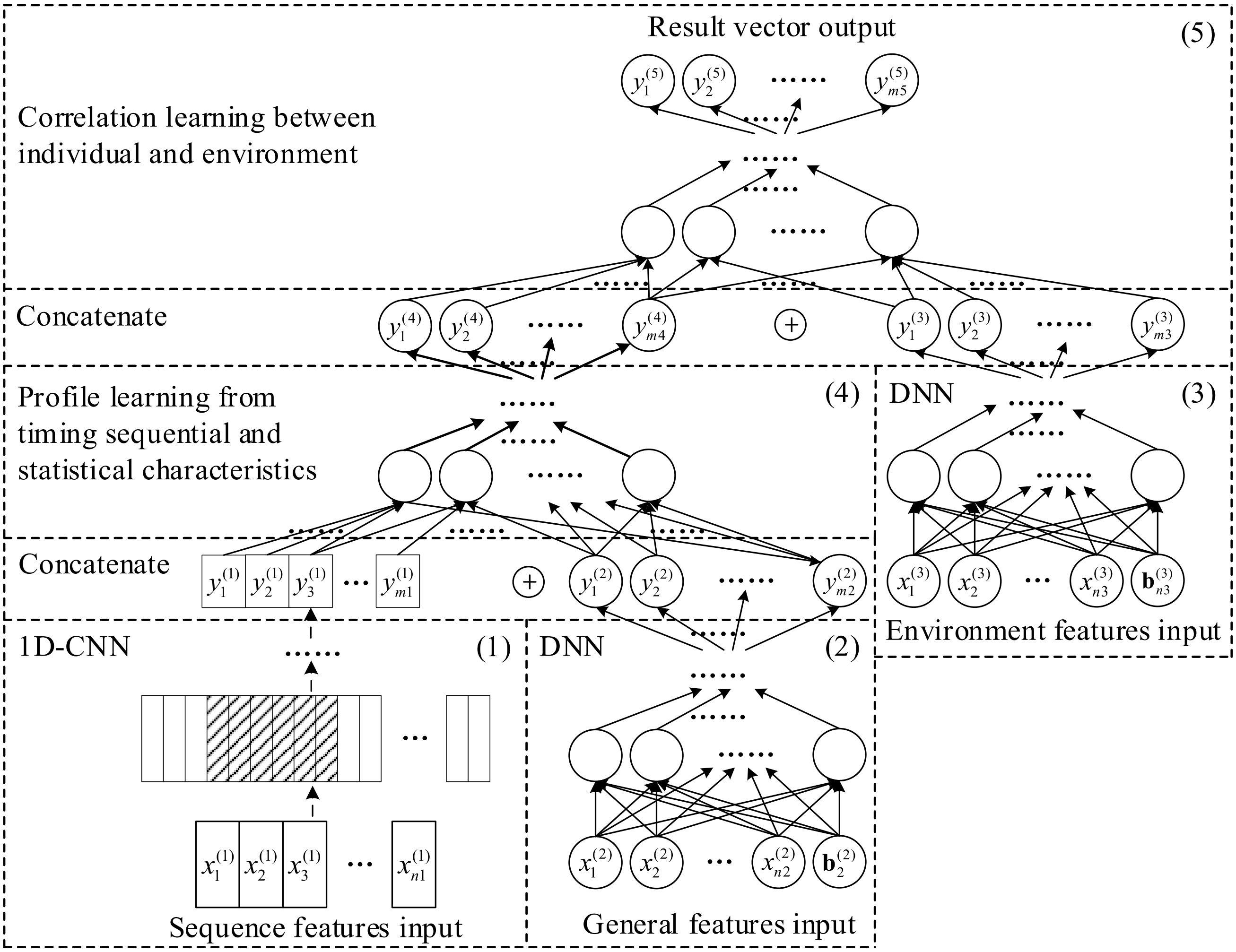

Result vector output
(5)
Correlation learning between individual and environment
$y_1^{(5)}$ $y_2^{(5)}$ ...... $y_{m5}^{(5)}$
Concatenate
$y_1^{(4)}$ $y_2^{(4)}$ ...... $y_{m4}^{(4)}$ ⊕ $y_1^{(3)}$ $y_2^{(3)}$ ...... $y_{m3}^{(3)}$
Profile learning from timing sequential and statistical characteristics
(4)
DNN
(3)
Concatenate
$y_1^{(1)}$ $y_2^{(1)}$ $y_3^{(1)}$ ... $y_{m1}^{(1)}$ ⊕ $y_1^{(2)}$ $y_2^{(2)}$ ...... $y_{m2}^{(2)}$
$x_1^{(3)}$ $x_2^{(3)}$ ... $x_{n3}^{(3)}$ $\mathbf{b}_{n3}^{(3)}$
Environment features input
1D-CNN
(1)
DNN
(2)
$x_1^{(1)}$ $x_2^{(1)}$ $x_3^{(1)}$ ... $x_{n1}^{(1)}$
Sequence features input
$x_1^{(2)}$ $x_2^{(2)}$ ... $x_{n2}^{(2)}$ $\mathbf{b}_2^{(2)}$
General features input

FIGURE 6. Hybrid neural network model based on 1D-CNN and DNN.

(1) The first part is a 1D-CNN. Characteristics of the packet sequence of a single flow are analyzed from the perspective of sequence by inputting packet sequence features of a single flow.
(2) The second part is a DNN that analyzes characteristics of a single flow from overall statistics viewpoint by inputting general statistical features of a single flow.
(3) The third part is a DNN, which analyzes current flow distribution characteristics by inputting environmental features of flows.
(4) The fourth part is a connection layer and a shallow neural network. The sequential and the statistical feature vectors outputted from the first two parts are concatenated and output to a shallow neural network, which comprehensively characterize single-flow characteristics from the perspective of sequential and statistical characteristics.
(5) The fifth part is also a connection layer and a shallow neural network. The single-flow integrated vector and the environmental characteristic vector outputted separately from the third and fourth parts are concatenated and output to a shallow neural network. It analyzes whether the flow is an abnormal flow or not from the perspective of single-flow characteristics and multi-flows distribution characteristics and outputs a prediction vector.

2) 1D-CNN

One-dimensional convolutional network (1D-CNN) has the advantage of being able to quickly learn sequential characteristics of data, which is composed of convolutional layers, pooling layers, and a global pooling layer.

The convolution operations in the CNN are based on discrete data. Fig.7 indicates that the convolution operation starts from one end of the input data to the other end, slides the fixed length per step sequentially and extracts a sequence of consecutive fixed lengths. By doing a dot product operation with a convolution kernel of the same size, the CNN obtains a new value, which is the eigenvalue of the corresponding position on the new sequence.

The role of the pooling layer is to downsample the feature sequence obtained by the convolution operation. In order to

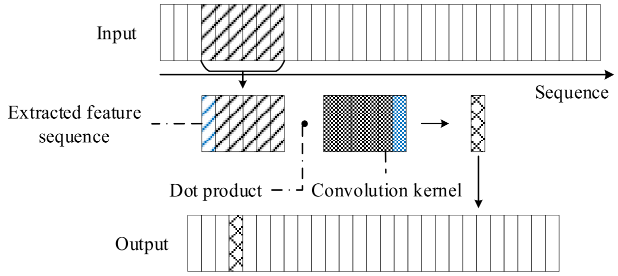


FIGURE 7. The processing of the 1D-CNN.

compress the feature sequence, the convolution kernel can be processed to a wider range of data faster. The hybrid neural network proposed in this study uses the maximum pooling layer. The principle is that the maximum value in the same depth in the pooled window is taken as the characteristic output of the depth.

Different from the general pooling layer, the global pooling layer flattens the high-dimension vector to the 1st dimension after compressing features. It means to convert a two-dimension tensor with size (n, m) to a vector with size $(n \times m)$ for subsequent network operations.

3) DNN

Deep neural network (DNN) can analyze the correlation between high-dimensional features of data, which consists of an input layer, an output layer and a number of hidden layers. The adjacent layers are connected by a full connection.

Fig.8 shows that neurons between adjacent layers of DNN are connected in a fully connected form. Where, $x = (x_1, x_2, x_3)^T$ is the input. Moreover, $\mathbf{y}_l$ and $y_l^{(i)}$ indicate the output vector of the layer l and the output vector of the i^{th} neuron in the layer l, respectively. Moreover, $\mathbf{u}_l$, $u_l^{(i)}$, $\mathbf{W}_l$ and $\mathbf{b}_l$ indicate the input vector of the layer l, the input vector of the i^{th} neuron in the layer l, the connection weight from the layer l-1 to the layer l and the connection offset from the layer l-1 to the layer l. The calculations are as shown in Eqs. (1) and (2).

$$\begin{cases} u_l^{(j)} = \sum_{i \in L_i} w_l^{(ji)} y_{l-1}^{(i)} + b_l^{(j)}, \quad l \geq 2 \\ y_l^{(j)} = f_l(u_l^{(j)}), \quad l \geq 2 \end{cases} \tag{1}$$

$$\boldsymbol{y}_l = f_l(\boldsymbol{u}_l) = f_l(\boldsymbol{W}_l \boldsymbol{y}_{l-1} + \boldsymbol{b}_l) \tag{2}$$

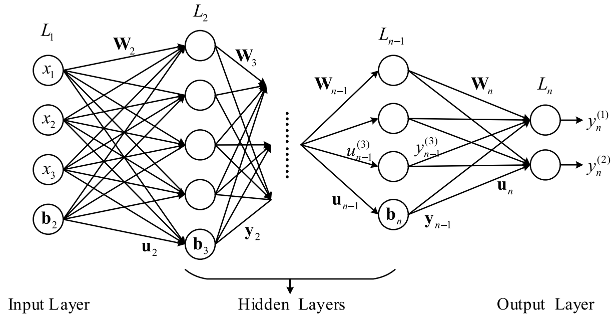


FIGURE 8. The construction of DNN.

where, $w_l^{(ji)}$ indicates the weight of the i^{th} neuron on the layer l-1 to the j^{th} neuron on the layer l, which is the element of the j^{th} row and the i^{th} column of $\boldsymbol{W}_l$. Moreover, $b_l^{(j)}$ represents the offset of the j^{th} neuron of layer l, which is the j^{th} element of $\boldsymbol{b}_l$.f_l $(\cdot)$ is the activation function of layer l.

4) DATA NORMALIZATION

Data normalization is often implemented in neural networks. After normalization, the mean value of new data is 0 and the standard deviation is 1, which makes the data distribution more concentrated and helps the gradient descent method to train network parameters more quickly. In Eq. (3), x_i and x_i' represent the i^{th} dimension feature of original data and the i^{th} dimension feature after data normalization, respectively. Moreover, $\mu(x)$ and $\sigma(x)$ represent the mean of original data and the standard deviation of original data, respectively. It should be indicated that the AD-H1CD performs data normalization for GSF and ENF.

$$x_i' = \frac{x_i - \mu(x)}{\sigma(x)} \tag{3}$$

IV. EXPERIMENTS AND EVALUATION

This section mainly analyzes performances of the AD-H1CD, including the evaluation of AD-H1CD parameters and verifying the effectiveness of the AD-H1CD hybrid structure. Moreover, it is intended to compare performances of different algorithms with that of the proposed method.

A. NEURAL NETWORK SETTING OF THE AD-H1CD

Table 4 shows hyper-parameter settings of the AD-H1CD scheme. First and second parameters of the Conv1D layer (One-dimension Convolution layer) indicate the depth of the convolution kernel and the size of the convolution window with the default value of one, respectively. Moreover, parameters of the MaxPool1D layer (one-dimension max pooling layer) and the dense layer represent the window size and the number of neurons, respectively.

Table 4 indicates that layers 1.1-1.7, 2.1-2.4 and 3.1-3.3 are used for processing the SPF, GSF and the ENF, respectively. Moreover, layers 4.1-4.4 are used for concatenating the learning vectors of SPF and GSF. Finally, layers 5.1-5.5 are utilized for concatenating the flow vector and the ENF vector, and generating the result vector.

The AD-H1CD scheme has two output modes, including binary classification and multi-classification. Item 5.5a of Table 4 indicates that the number of neurons of the output layer for the binary classification is one, and the activation function is sigmoid. On the other hand, item 5.5b of Table 4 indicates that the number of neurons of the output layer for the multi-class classification is the number of flow classes and the activation function is softmax.

In addition to the output layer, the activation function of the layers is set to Rectified Linear Unit (ReLU). The network is optimized by Root Mean Square Prop (RMSprop) with the batch size of 512.The binary cross entropy is used as the loss function for the binary classification, while categorical

TABLE 4. Hyperparameters setting of the hybrid neural network of AD-H1CD.

Num	Layer setting	Num	Layer setting
1.1	Conv1D (16, 5)	3.2	Dense (16)
1.2	MaxPool1D (2)	3.3	Dense (8)
1.3	Conv1D (32, 5)	4.1	Concatenate ()
1.4	MaxPool1D (3)	4.2	Dense (16)
1.5	Conv1D (64, 5)	4.3	Dense (16)
1.6	GlobalMaxPooling1D	4.4	Dense (8)
1.7	Dense (8)	5.1	Concatenate ()
2.1	Dense (64)	5.2	Dense (16)
2.2	Dense (64)	5.3	Dense (16)
2.3	Dense (32)	5.4	Dense (8)
2.4	Dense (8)	5.5a	Dense (1) + Sigmoid
3.1	Dense (16)	5.5b	Dense (m) + Softmax

cross entropy is used for the multi-classification. Moreover, it should be indicated that the optimization evaluation in the training process is accuracy.

B. EXPERIMENTAL SETTING

Two datasets of ISCX-IDS-2012[34] and CIC-IDS-2017[35] are utilized for experiments evaluation in the present study. The number of abnormal flows in these two datasets is much less than that of normal flows. When the two divide the training set and the test set in the same proportion, the performance of the trained model decrease, which originates from the class imbalance. Therefore, the training set of the normal and abnormal flows is divided in the present study in accordance with the actual distribution of datasets. Because the training number of both flows should be similar and the number of abnormal flows used for training should account for 80% of all abnormal flows in the dataset. The ratio of training and test data for normal flows is 0.05:0.95 for the ISCX-IDS-2012 dataset, while it is 0.1:0.9 for the CIC-IDS-2017 dataset. Moreover, the ratio of training and test data for abnormal flows is 8:2. Table 5 and 6 illustrate data partition for abovementioned datasets.

In experiments of parameter and effectiveness evaluation for the hybrid structure, only the training data is used for training and verification. The test data is only used for the final performance test and comparison with other algorithms. Furthermore, for the evaluation of parameters and hybrid structure, 10-fold cross-validation is used to ensure reliable results.

TABLE 5. Data partition for ISCX-IDS-2012 dataset (The number in parentheses is the number of flows).

Data type		Training		Test		All	
Normal		4.87% (105559)		92.59% (2005620)		97.46 % (2111179)	
Anomaly	Infiltrating	2.03% (43825)	0.37% (7941)	0.51% (10956)	0.09% (1985)	2.54% (54781)	0.46% (9926)
	HttpDos		0.13% (2730)		0.03% (682)		0.16% (3412)
	DDos		1.27% (27512)		0.32% (6878)		1.59% (34390)
	BFSSH		0.26% (5642)		0.07% (1411)		0.33% (7053)
All		6.90% (149384)		93.10% (2016576)		100% (2165960)	

TABLE 6. Data partition for CIC-IDS-2017 dataset (The number in parentheses is the number of flows).

Data type		Training		Test		All	
Normal		9.29% (161105)		83.57% (1449943)		92.86% (1611048)	
Anomaly	SSH Patator	5.71% (99065)	0.18% (3193)	1.43% (24767)	0.05% (798)	7.14% (123832)	0.23% (3991)
	FTP Patator		0.14% (2382)		0.03% (596)		0.17% (2978)
	Dos		1.6% (27674)		0.4% (6919)		1.99% (34593)
	Web attack		0.09% (1635)		0.02% (409)		0.12% (2044)
	DDos		0.1% (1783)		0.03% (446)		0.13% (2229)
	Bot		2.64% (45761)		0.66% (11440)		3.3% (57201)
	PortScan		0.96% (16637)		0.24% (4159)		1.2% (20796)
All		15.0% (260170)		85.0% (1474710)		100% (1734880)	

The device used in this experiment is MSI (GT63), which has 6 CPUs and each CPU is Intel(R) Core(TM) i7-8750H@2.2GHz, with the memory size of 16GB. The GPU is NVDIA GeForce GTX 1070. Keras is used to build the hybrid neural network.

C. EVALUATION METRIC

Four metrics of Loss, overall accuracy (OA-ACC), detection rate (DR) and false alarm rate (FAR) are used to evaluate the proposed method in experiments. Since the AD-H1CD scheme can achieve two-class or fine-grained multi-class anomaly detection, this study uses a unified formal description to express these four indicators, as shown in Eqs. (4)-(7). The cross-entropy calculation, which can be used as the loss function, is expressed in the form below:

$$\text{Loss} = -\frac{1}{N} \cdot \begin{cases} \sum_{j=1}^{N} \sum_{i=1}^{K} (z_j^{(i)} ln y_j^{(i)}), & K > 2 \\ \sum_{j=1}^{N} [z_j ln y_j + \left(1 - z_j\right) ln(1 - y_j)], & K = 2 \end{cases} \tag{4}$$

where N and K are the number of the predicted data and the size of the label set, respectively. Moreover, y_j and z_j denote the predicted label result of j^{th} data and the true label, respectively. When K is 2, y_j and z_j are real parameters between 0 to 1. Otherwise, when K exceeds 2, the label value should be done binarization for the multi-class classification. For example, when K is 3, the 3rd label value should be transformed as $z_j = (0, 0, 1)$. Therefore, $z_j^{(i)}$ and $y_j^{(i)}$ are i^{th} component of j^{th} data, which indicates the probability of i^{th} label of the j^{th} data.

$$\text{OA} - \text{ACC} = \frac{\sum_{i=1}^{K} TP_i}{N} \tag{5}$$

$$\text{DR}_i = \frac{TP_i}{TP_i + FN_i} \tag{6}$$

$$\text{FAR}_i = \frac{FP_i}{TN_i + FP_i} \tag{7}$$

Parameters of the abovementioned equations are defined as follows:

TP_i indicates the number of the i^{th}-class data that is correctly detected.

TN_i indicates the number of the not i^{th}-class data that is correctly detected.

FP_i indicates the number of the not i^{th}-class data that is incorrectly detected as ith-class data.

FN_i indicates the number of the i^{th}-class data that is incorrectly detected.

Similar to Eq. (4), N and K are the number of the predicted data and the number of labels, respectively.

D. PARAMETERS EVALUATION

This section analyzes three parameters of epoch, n and α, which indicate the training round number, the number of data packets for analysis of SPF and GSF and the parameter of sliding window length, respectively.

1) EPOCH

Fig.9 shows that the present study obtains epoch from 1 to 30and compares the Loss and OA-ACC in the case of binary classification and multi-classification on the ISCX-IDS-2012 and the CIC-IDS-2017 datasets. In terms of loss, after 1 epoch, the Loss remains stable with the value less than 0.03 and it slowly decreases with the growth of epoch. The smaller the Loss, the better the system performance. In terms of OA-ACC, as the epoch gradually increases, the OA-ACC of the training and verification increase without over-fitting. After the epoch reaches 15, the growth rate of OA-ACC slows down. Moreover, the trend of the training curve and the verification curve are basically the same, indicating that the training has not been over-fitting. It is observed that the training process of the AD-H1CD scheme is robust and even if the epoch reaches 30, there is no over-fitting. In order to make balance between the model accuracy and training time consumption, a suitable epoch value is 15.

2) *n*

Fig. 10 shows the number of packets analyzed, which is named n, obtained from 10 to 100 and compares OA-ACC values in the case of the binary classification and multi-classification on the ISCX-IDS-2012 and the CIC-IDS-2017 datasets. It should be indicated that the trend of the variation of the OA-ACC is different for two datasets. In general, the trend of change with n is relatively flat and peak points are not the same in 4 subgraphs, however all of them are in postmedian, that is, 70-100. It seems that AD-H1CD is not sensitive to the change of n and OA-ACC has no obvious change trend with the growth of n. In order to minimize resource consumption, this study considers the optimal n value to be 90.

3) α

Fig. 11 indicates that the flow sliding window parameter α obtains the value from 1 to 10 and compares OA-ACC values in the case of the binary classification and multi-classification on the ISCX-IDS-2012 and the CIC-IDS-2017 datasets. It should be indicated that changes of the OA-ACC are different for different datasets and the value of α has a greater impact on the CIC-IDS-2017 dataset. The OA-ACC gradually increases as the value of α increases, however finally there is a downward trend in the CIC-IDS-2017 dataset, while the change in the ISCX-IDS-2012 dataset is flat and there is almost no significant change. Based on the four subgraphs, the best values are concentrated in the interval of 8-10, and the optimal value can be 9.

E. EFFECTIVENESS EVALUATION OF THE HYBRID CONSTRUCTION

Based on advantages of the hybrid structure, the AD-H1CD method learns sequential, general statistical and environmental characteristics of network flows. This section analyzes

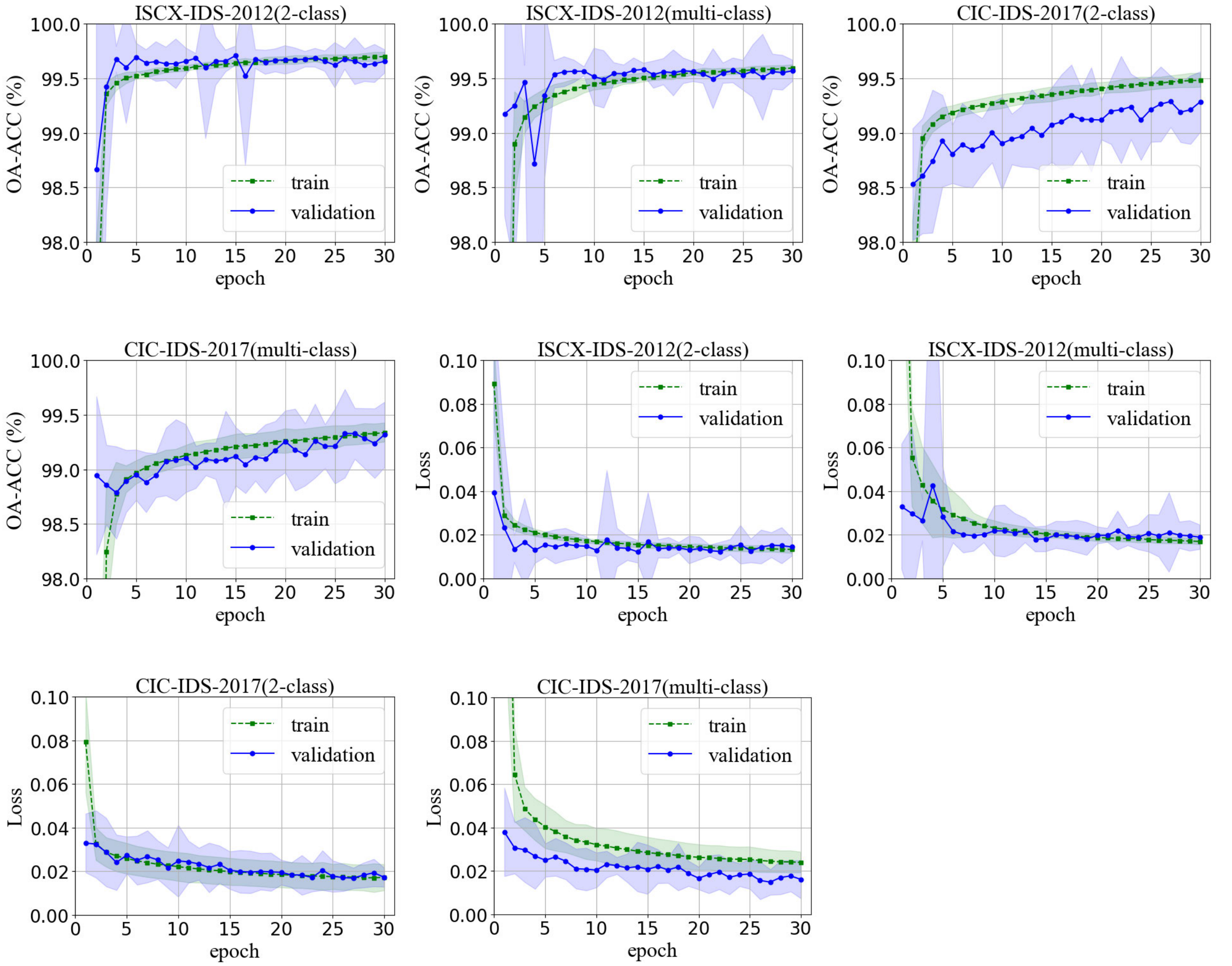


FIGURE 9. Comparison of OA-ACC and loss under different epoch values on two datasets.

TABLE 7. Performances comparison of substructures of AD-H1CD on ISCX-IDS-2012 dataset (%).

Construction	Normal		Infiltrating		HttpDos		DDos		BFSSH		OA-ACC
	DR	FAR	DR	FAR	DR	FAR	DR	FAR	DR	FAR	
1D-CNN(SPF)	99.59	0.67	95.84	1.68	53.04	9.67	39.78	0.11	99.68	0.07	87.53
DNN(GSF)	99.45	0.97	91.05	0.32	81.31	0.36	99.56	0.35	99.64	0.07	98.70
DNN (ENF)	97.63	0.45	95.86	2.66	87.83	0.06	98.92	1.67	30.46	0.35	95.06
1D-CNN(SPF) + DNN(GSF)	99.62	0.97	95.21	0.48	75.45	0.12	99.52	0.22	99.29	0.04	98.91
AD-H1CD	99.62	0.18	99.11	0.03	98.90	0.03	99.67	0.37	99.99	0.06	99.56

note: A(B) means processing of B based on A; "+" is the concatenate operation of two model.

the practical effectiveness of this hybrid structure in anomaly network flow detection.

Tables 7 and 8 illustrate the performance comparisons of substructures in the AD-H1CD method in the ISCX-IDS-2012 and the CIC-IDS-2017 datasets. It is observed that hybrid structures of the AD-H1CD scheme are effective. Hybrid structures of SPF and GSF processing (1D-CNN(SPF) + DNN(GSF)) and the complete hybrid structure show more accurate detection results than single-type features processing structure. Separate processing models show obvious disadvantages. It is observed that the structure for processing with the SPF has a poor effect on

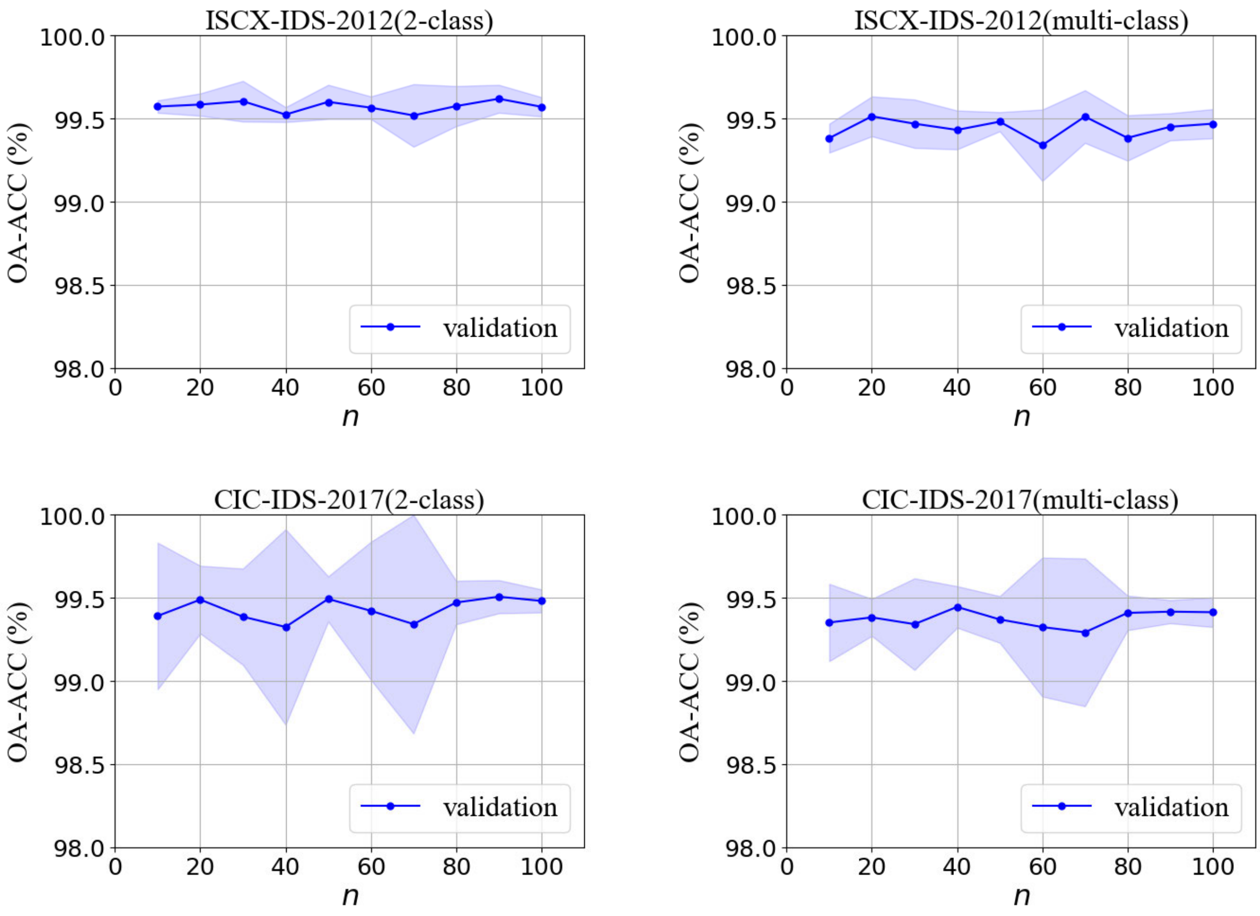


FIGURE 10. Comparison of OA-ACC under different *n* values for two datasets.

the distributed anomaly detection, including DDos and httpDos. Moreover, the structure for processing with ENF has a poor effect on the anomaly detection based on the single-flow, such as Bruce Force. It should be indicated that the AD-H1CD scheme not only maintains the advantages of each sub-network structure, including the detection of Bruce Force by SPF processing structure and the detection of DDos attack by ENF, but also it has a good DR for some abnormalities such as HttpDos that cannot be detected by structures based on analysis of single-type features through the fusion of characteristics. However, there are some anomalies that single structures get good DRs, but AD-H1CD does not, such as web attack. However, the hybrid structure of the AD-H1CD combines the most advantages of structures based on processing single-type features and improving the DR of some anomalies by using the correlation between features, which indicates that the hybrid construction proposed in the present study is effective.

F. TEST RESULTS AND COMPARISON

Tables 9 and 10 show experimental results of the AD-H1CD scheme of two datasets, including binary classification and multi-classification results. Moreover, tables 11-15 compares the performances of the AD-H1CD on test data with state-of-the-art algorithms. Tables include the binary classification performance comparisons, multi-classification performance comparison and time consumption comparison on the ISCX-IDS-2012 and the CIC-IDS-2017 datasets. Since the neural network is greatly affected by initial parameters, experiments are repeated 10 times with the same database in order to obtain more objective experimental results.

The AD-H1CD scheme shows stable and high anomaly DR on both datasets, especially for multi-flow implementations such as port scan and DDos, which the DR reaches 99.8% and 99.9%, respectively. It should be indicated that the DR of the SSH Patator maintains 100% in 10 times tests and DRs of other anomalies reach 99% except Web attack and Bot. DR of the web attack is 87.38%, which is clearly lower than that for other DRs. The possible description for this is that web attack contains different kinds of attacks and their characteristics are not uniform or the features of SPF, GSF, ENF do not characterize web attack well. In fact, web attack could be implemented through the single flow or multi flows. Therefore, the corresponding statistical characteristics are hard to be mined, which leads to the result of low detection rate. Moreover, it is observed that the standard deviation of

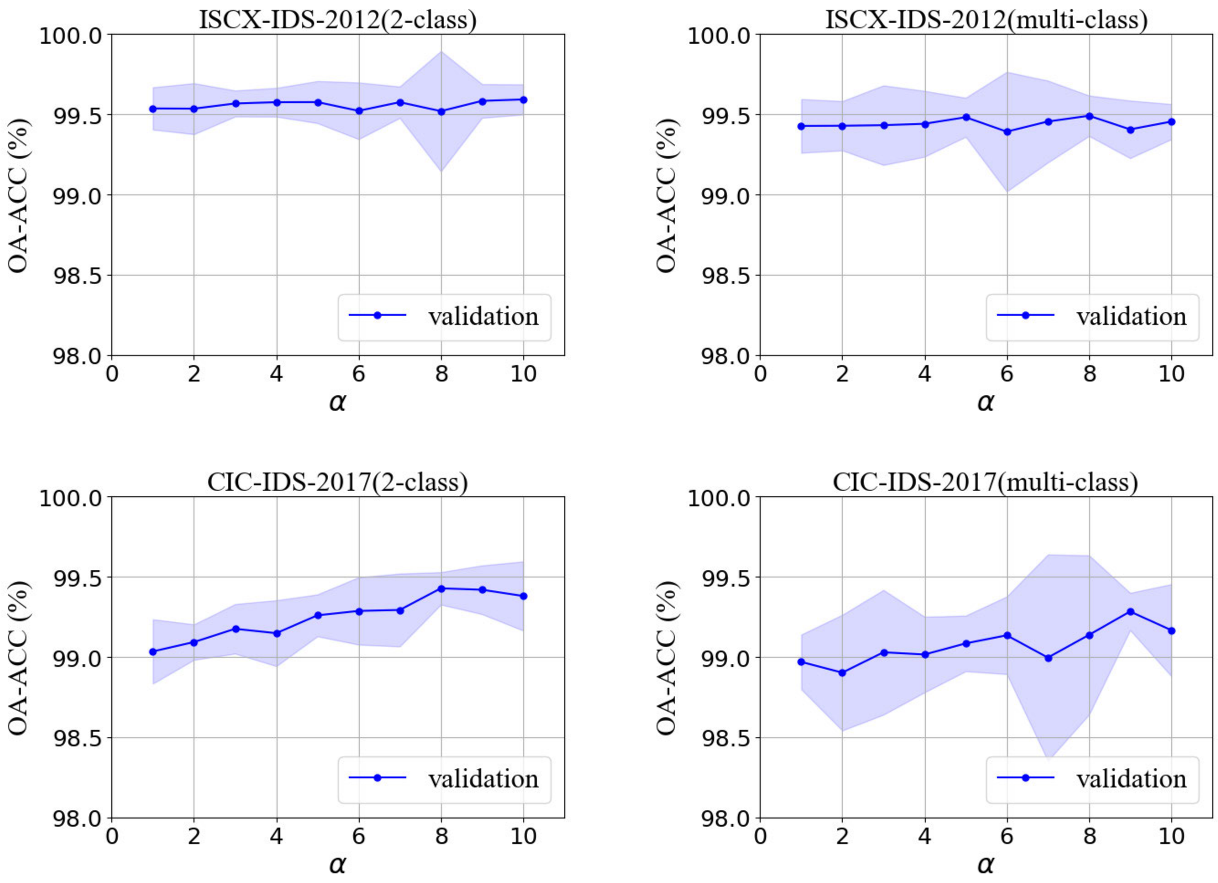


FIGURE 11. Comparison of OA-ACC under different α values of two datasets.

TABLE 8. Performances comparison of substructures of AD-H1CD on CIC-IDS-2017 dataset (%).

Construction	Normal		SSH Patator		FTP Patator		Dos	
	DR	FAR	DR	FAR	DR	FAR	DR	FAR
1D-CNN(SPF)	98.41	12.85	99.74	0.00	57.14	0.00	76.10	4.14
DNN(GSF)	98.33	0.36	99.99	0.00	98.73	0.00	99.74	0.03
DNN (ENF)	96.94	11.49	0.00	0.00	0.00	0.00	89.90	2.20
1D-CNN(SPF) + DNN(GSF)	98.46	0.17	99.99	0.00	99.99	0.00	99.67	0.00
AD-H1CD	99.68	0.58	99.99	0.00	99.57	0.00	99.89	0.01

Construction	Web attack		Bot		PortScan		DDos		OA-ACC
	DR	FAR	DR	FAR	DR	FAR	DR	FAR	
1D-CNN(SPF)	71.34	0.26	73.25	0.17	99.67	0.95	19.93	0.03	90.53
DNN(GSF)	97.56	0.32	94.94	0.14	99.54	0.69	99.93	0.00	98.79
DNN (ENF)	0.00	0.00	2.47	0.01	97.5	1.23	73.81	1.86	91.48
1D-CNN(SPF) + DNN(GSF)	95.73	0.28	96.62	0.09	99.89	0.72	99.63	0.03	98.92
AD-H1CD	71.34	0.04	95.50	0.06	99.84	0.09	99.99	0.02	99.55

note: A(B) means processing of B based on A; "+" is the concatenate operation of two model.

Web attack and Bot is higher than 12%, while others are less than 1%. In 10 times test, the maximum DR of Web attack is 95.59% and Bot is 97.09%. It seems that the training of these two anomalies detection is under fitting, however good performances prove that AD-H1CD can train a model to detect these two anomalies.

TABLE 9. Multi-classification performances of AD-H1CD on test sets (%).

Dataset	Data class	DR	FAR	OA-ACC
ISCX-IDS-2012	Normal	99.61±0.06	0.49±0.14	99.61±0.06
	Infiltrating	98.15±0.75	0.03±0.04	
	HttpDos	99.03±0.47	0.01±0.00	
	DDos	99.68±0.27	0.31±0.04	
	BFSSH	99.22±0.84	0.02±0.03	
CIC-IDS-2017	Normal	99.35±0.18	0.47±0.41	99.35±0.17
	SSH Patator	100.0±0.00	0.00±0.00	
	FTP Patator	98.85±1.02	0.00±0.00	
	Dos	99.63±0.58	0.00±0.00	
	Web attack	87.38±13.7	0.32±0.19	
	Bot	94.48±12.8	0.13±0.06	
	PortScan	99.82±0.12	0.16±0.06	
	DDos	99.86±0.05	0.00±0.00	

TABLE 10. Binary classification performances of AD-H1CD on test sets (%).

Dataset	Data class	DR	FAR	OA-ACC
ISCX-IDS-2012	Normal	99.58±0.05	0.31±0.14	99.58±0.04
	Anomaly	99.68±0.14	0.41±0.05	
CIC-IDS-2017	Normal	99.58±0.17	0.71±0.31	99.57±0.16
	Anomaly	99.28±0.31	0.41±0.17	

TABLE 11. Binary classification performance comparison between AD-H1CD and other algorithms on the ISCX-IDS-2012 dataset (%).

Method	Anomaly		OA-ACC
	DR	FAR	
Pektaset al.(2019)[29]	99.08	2.27	99.09
HAST-IDS(2017)[31]	96.96	0.02	99.89
ALL-AGL(2013)[36]	93.22	0.3	95.35
KMC+NBC(2013)[37]	99.70	2.2	99.0
AMGA2-NB(2013)[38]	92.7	7.0	94.5
MHCVF(2016)[39]	68.2	0.03	99.5
Tan et al.(2015)[40]	90.04	7.92	90.12
KMC-D+NBC(2016)[41]	99.37	0.03	99.52
AD-H1CD	99.68	0.41	99.58

TABLE 12. Multi-classification performance comparison between AD-H1CD and other algorithms on the ISCX-IDS-2012 dataset (%).

Method	Normal	Infiltrating	Httpdos	DDos	BFSSH
	DR	DR	DR	DR	DR
HAST-IDS(2017)[31]	**99.97**	96.21	92.88	97.95	97.09
MHCVF(2016)[39]	99.96	19.7	79.02	66.28	99.06
AD-H1CD	99.61	**98.15**	**99.03**	**99.68**	**99.22**

TABLE 13. Binary classification performance comparison between AD-H1CD and other algorithms on the CIC-IDS-2017 dataset (%).

Method	Anomaly		OA-ACC
	DR	FAR	
Pektas et al. (2019)[29]	98.83	2.64	98.87
DeepDetect(2019)[30]	99.70	1.88	98.69
AD-H1CD	99.28	**0.41**	**99.57**

TABLE 14. Multi-classification performance comparison between AD-H1CD and other algorithms on the CIC-IDS-2017 dataset (%).

	Normal		SSH Patator		FTP Patator		Dos	
	DR	FAR	DR	FAR	DR	FAR	DR	FAR
Han et al. (2019)[28]	**99.50**	2.76	95.70	0.17	97.71	0.12	**100**	0.00
AD-H1CD	99.35	**0.47**	**100.0**	**0.00**	**98.85**	**0.00**	99.63	**0.00**
	Web attack		Bot		Port scan		DDos	
	DR	FAR	DR	FAR	DR	FAR	DR	FAR
Han et al. (2019)[28]	**96.02**	**0.02**	88.31	**0.07**	96.93	**0.03**	98.00	0.04
AD-H1CD	87.38	0.32	**94.48**	0.13	**99.82**	0.16	**99.86**	**0.00**

Comparison of the AD-H1CD scheme with other algorithms shows that DRs for anomalies of the AD-H1CD are very prominent and DRs for all kinds of anomalies are almost the highest compared to other algorithms. Although it is not always the highest of all algorithms, however the OA-ACC and FRs of the AD-H1CD is much better than the algorithm with the highest DR. Meanwhile, compared with other algorithms that perform well in OA-ACC or FAR, AD-H1CD has a more prominent advantage in anomaly detection.

In order to compare the time consumption of the proposed method with that of other methods, time consumptions of different methods are presented in Table 15. However, the direct comparison may impose remarkable error to the evaluation. Because time consumptions are not obtained from the same platform. Although it would be better if the methods run on

TABLE 15. Time consumption comparison between AD-H1CD and other algorithms.

	Train	Test	Train size	Test size
HAST-IDS(2017)[31]	58min	1.7min	2,466,929	1,099,731
MHCVF(2016)[39]	240min	35min	553,392	368,928
AD-H1CD	**2min**	**2.4min**	105,559	2,005,620

the same hardware, it is a great challenge to obtain the source codes of other methods. Therefore, the data are directly chosen from the corresponding literatures. Table 15 shows that application of the AD-H1CD significantly reduces the time consumption. From the information of Keras, the time consumption of training per step is 24us. It is the time of neural network training based on Back Propagation (BP) algorithm per step with a batch of data. The value setting of a batch is 512. On the one hand, the AD-H1CD does not use recurrent neural network such as LSTM that consumes more resources, on the other hand, selected features presented in this study are easy to be extracted and analyzed. Combined with the anomaly detection performance of the AD-H1CD and time consumption, it can be concluded that the AD-H1CD is a good network flow anomaly detection method, which has strong practicability.

G. DISCUSSION

Unlike other anomaly detection methods, the AD-H1CD method extracts and analyzes multi-type flow features so that it can theoretically analyze different anomalies from a more comprehensive perspective. Moreover, it captures the patterns of the anomalous flows by analyzing the characteristics in terms of sequential, statistical and environmental characteristics. The effectiveness of the proposed method is proved through the experimental results. It is found that the feature dimension of the AD-H1CD is so high that it is not easily overfit, even when the epoch is greater than 10, the accuracy of the model is still improving. Meanwhile, it is robust for its insensitivity in changes of parameters, such as n and α. Moreover, this paper has finished the experiments that trains the model on one dataset and tests it on another dataset. However, the results are not satisfactory. Although the detection rates of normal flows are higher than 99%, the detection rates of abnormal flows are lower than 40%. Maybe it is because the characteristics of normal flows are similar in different networks, while anomalies are not. Therefore, it is difficult for the model trained on one network traffic to detect the anomalies of other networks. In general, AD-H1CD model trained on a special network can show good performances on this network both in excellent anomaly detection effect and low time consumption, but the model is not generalized enough that it could not be used in other networks to detect anomalies.

V. CONCLUSION

Considering the security status of LLANs, the present study proposes an anomaly detection method for network flows (AD-H1CD). The proposed method is based on the hybrid neural network and is comprised of the 1D-CNN and the DNN schemes. By extracting multi-type features of network flows and constructing a hybrid neural network, network flows are comprehensively analyzed, and an efficient, accurate and fine-grained network scheme for the anomaly detection is proposed. Compared with the conventional anomaly detection algorithms, the detection rate of the AD-H1CD algorithm for anomalies in the multi-classification has been significantly improved. Moreover, the training and testing time of the model is shorter. However, the generalization of this method should be improved. To solve this problem, it is intended to design a set of generalized features in the future investigations.

APPENDIX

ABBREVIATION NOUNS

Abbreviation	Full name
AD-H1CD	Anomaly Detection for network flows based on Hybrid neural network comprised of 1D-CNN and DNN
DNN	Deep Neural Network
CNN	Convolution Network
1D-CNN	One-dimensional Convolution Network
SPF	Sequence Packets Features
GSF	General Statistical Features
ENF	Environmental Features
IAT	Packet Inter-arrival Time
Conv1D	One-dimensional Convolution layer
LLAN	Large-scale Local Area Network

REFERENCES

[1] G. Fernandes, Jr., J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, Jr., "A comprehensive survey on network anomaly detection," *Telecommun. Syst.*, vol. 70, no. 3, pp. 447–489, Mar. 2019. doi: 10.1007/s11235-018-0475-8.

[2] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Gener. Comput. Syst.*, vol. 55, pp. 278–288, Feb. 2016. doi: 10.1016/j.future.2015.01.001.

[3] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019. doi: 10.1016/j.jnca.2018.12.006.

[4] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2018. doi: 10.1109/ACCESS.2018.2886457.

[5] T. Sun and H. Tian, "Anomaly detection by diffusion wavelet-based analysis on traffic matrix," in *Proc. 6th Int. Symp. Parallel Archit., Algorithms Program.*, Jul. 2014, pp. 148–151. doi: 10.1109/paap.2014.62.

[6] X. Jing, Z. Yan, X. Jiang, and W. Pedrycz, "Network traffic fusion and analysis against DDoS flooding attacks with a novel reversible sketch," *Inf. Fusion*, vol. 51, pp. 100–113, Nov. 2019. doi: 10.1016/j.inffus.2018.10.013.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009, Art. no. 15. doi: 10.1145/1541880.1541882.

[8] E. Hou, Y. Yılmaz, and A. O. Hero, "Anomaly detection in partially observed traffic networks," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1461–1476, Mar. 2019. doi: 10.1109/TSP.2019.2892026.

[9] J. David and C. Thomas, "Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic," *Comput. Secur.*, vol. 82, pp. 284–295, May 2019. doi: 10.1016/j.cose.2019.01.002.

[10] R. Paffenroth, K. Kay, and L. Servi, "Robust PCA for anomaly detection in cyber networks," Jan. 2018, *arXiv:1801.01571*. [Online]. Available: https://arxiv.org/abs/1801.01571

[11] A. Kind, M. P. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *IEEE Trans. Netw. Service Manag.*, vol. 6, no. 2, pp. 110–121, Jun. 2009. doi: 10.1109/TNSM.2009.090604.

[12] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," in *Proc. 8th ACM SIGCOMM Conf. Internet Meas.*, Oct. 2008, pp. 151–156. doi: 10.1145/1452520.1452539.

[13] M. V. Mahoney, ''Network traffic anomaly detection based on packet bytes,'' in *Proc. ACM Symp. Appl. Comput.*, Mar. 2003, pp. 346–350. doi: 10.1145/952532.952601.

[14] G. Thatte, U. Mitra, and J. Heidemann, ''Parametric methods for anomaly detection in aggregate traffic,'' *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 512–525, Apr. 2011. doi: 10.1109/TNET.2010.2070845.

[15] M. Javed, A. B. Ashfaq, M. Z. Shafiq, and S. A. Khayam, ''On the inefficient use of entropy for anomaly detection,'' in *Proc. Int. Symp. Recent Adv. Intrusion Detection*, Sep. 2009, pp. 369–370. doi: 10.1007/978-3-642-04342-0_28.

[16] H. Ringberg, A. Soule, J. Rexford, and C. Diot, ''Sensitivity of PCA for traffic anomaly detection,'' *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 35, no. 1, pp. 109–120, Jun. 2007. doi: 10.1145/1254882.1254895.

[17] K. Tamura and K. Matsuura, ''Improvement of anomaly detection performance using packet flow regularity in industrial control networks,'' *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E102-A, no. 1, pp. 65–73, 2019.

[18] J. Ting, R. Field, A. Fisher, and T. Bauer, ''Compression analytics for classification and anomaly detection within network communication,'' *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1366–1376, May 2019. doi: 10.1109/TIFS.2018.2878172.

[19] J. Zhang, R. Gardner, and I. Vukotic, ''Anomaly detection in wide area network meshes using two machine learning algorithms,'' *Future Gener. Comput. Syst.*, vol. 93, pp. 418–426, Apr. 2019. doi: 10.1016/j.future.2018.07.023.

[20] W. Alhakami, A. ALharbi, S. Bourouis, R. Alroobaea, and N. Bouguila, ''Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection,'' *IEEE Access*, vol. 7, pp. 52181–52190, 2019. doi: 10.1109/ACCESS.2019.2912115.

[21] X. Miao, Y. Liu, H. Zhao, and C. Li, ''Distributed online one-class support vector machine for anomaly detection over networks,'' *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1475–1488, Apr. 2019. doi: 10.1109/TCYB.2018.2804940.

[22] Y. Djenouri, A. Belhadi, J. C.-W. Lin, and A. Cano, ''Adapted K-nearest neighbors for detecting anomalies on spatio-temporal traffic flow,'' *IEEE Access*, vol. 7, pp. 10015–10027, 2019. doi: 10.1109/ACCESS.2019.2891933.

[23] Y. Chen and S. Li, ''A lightweight anomaly detection method based on SVDD for wireless sensor networks,'' *Wireless Pers. Commun.*, vol. 105, pp. 1235–1256, Apr. 2019. doi: 10.1007/s11277-019-06143-1.

[24] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, ''Network traffic anomaly detection using recurrent neural networks,'' Mar. 2018, *arXiv:1803.10769*. [Online]. Available: https://arxiv.org/abs/1803.10769

[25] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, ''GEE: A gradient-based explainable variational autoencoder for network anomaly detection,'' Mar. 2019, *arXiv:1903.06661*. [Online]. Available: https://arxiv.org/abs/1903.06661

[26] M. R. Moore and J. M. Vann, ''Anomaly detection of cyber physical network data using 2D images,'' in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–5.

[27] S. Ger and D. Klabjan, ''Autoencoders and generative adversarial networks for anomaly detection for sequences,'' Sep. 2019, *arXiv:1901.02514*. [Online]. Available: https://arxiv.org/abs/1901.02514

[28] L. Han, Y. Sheng, and X. Zeng, ''A packet-length-adjustable attention model based on bytes embedding using flow-wgan for smart cybersecurity,'' *IEEE Access*, vol. 7, pp. 82913–82926, 2019.

[29] A. Pektaş and T. Acarman, ''A deep learning method to detect network intrusion through flow-based features,'' *Int. J. Netw. Manage.*, vol. 29, May/Jun. 2019, Art. no. e2050. doi: 10.1002/nem.2050.

[30] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, ''DeepDetect: Detection of distributed denial of service attacks using deep learning,'' *Comput. J.*, Jan. 2019. doi: 10.1093/comjnl/bxz064.

[31] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, ''HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection,'' *IEEE Access*, vol. 6, pp. 1792–1806, 2017. doi: 10.1109/ACCESS.2017.2780250.

[32] Y. Zeng, H. Gu, W. Wei, and Y. Guo, ''*Deep-full-range*: A deep learningbased network encrypted trafficclassification and intrusiondetection framework,'' *IEEE Access*, vol. 7, pp. 45182–45190, 2019. doi: 10.1109/ACCESS.2019.2908225.

[33] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, ''Characterization of encrypted and VPN traffic using time-related features,'' in *Proc. 2nd Int. Conf. Inf. Syst. Secur. Privacy*, Rome, Italy, Feb. 2016, pp. 407–414.

[34] 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. doi: 10.1016/j.cose.2011.12.012.

[35] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ''Toward generating a new intrusion detection dataset and intrusion traffic characterization,'' in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, Lisbon, Portugal, Jan. 2018, pp. 108–116.

[36] H. Sallay, A. Ammar, M. B. Saad, and S. Bourouis, ''A real time adaptive intrusion detection alert classifier for high speed networks,'' in *Proc. IEEE 12th Int. Symp. Netw. Comput. Appl.*, Cambridge, MA, USA, Aug. 2013, pp. 73–80.

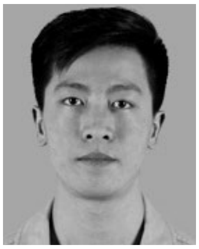
[37] W. Yassin, N. I. Udzir, Z. Muda, and M. N. Sulaiman, ''Anomaly-based intrusion detection through K-means clustering andnaives Bayes classification,'' in *Proc. 4th Int. Conf. Comput. Inform. (ICOCI)*, Aug. 2013, pp. 298–303.

[38] G. Kumar and K. Kumar, ''Design of an evolutionary approach for intrusion detection,'' *Sci. World J.*, vol. 2013, Sep. 2013, Art. no. 962185.

[39] A. Akyol, M. Hacibeyoğlu, and B. Karlik, ''Design of multilevel hybrid classifier with variant feature sets for intrusion detection system,'' *IEICE Trans. Inf. Syst.*, vols. E99-D, no. 7, pp. 1810–1821, Jul. 2016.

[40] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, ''Detection of denial-of-service attacks based on computer vision techniques,'' *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2519–2533, Sep. 2015.

[41] H. M. Tahir, A. M. Said, N. H. Osman, N. H. Zakaria, P. N. M. Sabri, and N. Katuk, ''Oving K-means clustering using discretization technique in network intrusion detection system,'' in *Proc. 3rd Int. Conf. Comput. Inf. Sci. (ICCOINS)*, Aug. 2016, pp. 248–252. doi: 10.1109/ICCOINS.2016.7783222.



CHENCHENG MA received the B.S. degree from the Zhengzhou Science and Technology Institute, Zhengzhou, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include network security and machine learning.

XUEHUI DU received the Ph.D. degree from the Zhengzhou Science and Technology Institute, Zhengzhou, China, in 2012, where she is currently a Professor. Her research interests include cloud computing and big data security.

LIFENG CAO received the Ph.D. degree from the Zhengzhou Science and Technology Institute, Zhengzhou, China, in 2013, where he is currently an Associate Professor. His research interests include cloud computing and information security.