DOE-DTL: A ML-Utilized System Combined With PDP for Detection and Mitigation of DLDoS Attack

Dan Tang, Xinmeng Li, Pei Tan, Keqin Li, *Fellow, IEEE*, Zheng Qin, *Associate Member, IEEE*, and Jiliang Zhang, *Senior Member, IEEE*

***Abstract*—Software-Defined Network (SDN) revolutionizes traditional network structures by isolating the data plane and the control plane, which offers greater flexibility in managing network resources. Nevertheless, SDN remains vulnerable to certain threats inherited from the traditional network, including Distributed Low-rate Denial-of-Service (DLDoS) attack. This attack is more subtle and harder to detect than traditional Distributed Denial-of-Service (DDoS) attacks, because it employs a lower average attack rate. We design a real-time detection and mitigation system named DOE-DTL specifical for the DLDoS attack in SDN. For the DLDoS attack detection, we utilize Machine-Learning (ML) methods to construct a detection model and introduce it in DOE-DTL. In the construction, we leverage Extreme Learning Machine (ELM) and make a dual optimization using Whale Optimization Algorithm (WOA). For the DLDoS attack mitigation, we use double thresholds to determine the attack sources and make corresponding mitigation rules. DOE-DTL innovatively combines the Programmable Data Plane (PDP) in detection and mitigation, shifting some control plane tasks to the data plane. Performance assessments reveal that DOE-DTL ensures fast, accurate attack identification and low-latency mitigation while maintaining low resource usage.**

***Index Terms*—Attack detection and mitigation, DLDoS attack, dual optimization, double thresholds, ELM, PDP, WOA.**

I. Introduction

SOFTWARE-DEFINED Network (SDN) largely improves the convenience compared to the traditional network, with the decoupled control and data planes, SDN can update and deploy network functions more flexibly [1]. However, since SDN continues to use communication protocols in the traditional network architecture, the attacks based on the mechanical principles of these protocols are still applicable in it, such as Distributed Low-rate Denial-of-Service (DLDoS) attack. DLDoS attack exploits the TCP congestion control mechanism. It is a multi-source Low-rate Denial-of-Service (LDoS) attack [2], each attack source periodically generates high-intensity traffic pulses to preempt the TCP bandwidth, causing the TCP congestion control mechanism continuously makes adjustment for the network traffic transmission and the normal TCP traffic transmission is affected. DLDoS attack can adversely affect the quality of network service with lower attack cost. DLDoS attack demonstrates higher concealment than traditional Distributed Denial-of-Service (DDoS) attacks with the lower average attack rate [3], making it particularly dangerous for SDN.

At present, researches on the DDoS attack confrontation in SDN mainly focus on flooding attacks, and there are not many methods specifically targeting DLDoS attack. Moreover, almost all the DLDoS attack detection and mitigation schemes that have been proposed are designed to be deployed entirely on the controller. Because before the Programmable Data Plane (PDP) [4] is proposed, the data plane could not be customized as required, researchers are more accustomed to exploiting the control plane programmability to deploy policies and perform centralized control. The methods that need to be completely deployed on the control plane create substantial processing burdens, with the data-control plane interactions producing inevitable time and resource costs. After the PDP appears, SDN can customize the data packet processing and forwarding logic on it to realize some detection and mitigation functions. And the adjustment of the functions is very convenient due to the flexible programmability of the PDP.

We propose DOE-DTL, a real-time detection and mitigation system combined with PDP for the DLDoS attack in SDN. DOE-DTL offloads part of the work from the control plane to the PDP, which increases the flexibility and reduces the pressure of the control plane as well as the time and resource consumption to a certain extent. It has seven modules: Data Statistics, Feature Extraction, Traffic Monitoring, Attack Detection, Suspicious IP Location, Suspicious IP List Maintenance, and Mitigation Rule Deployment module. The Traffic Monitoring module carries out coarse-grained pre-detection, and the Attack Detection module uses a DLDoS attack detection model to detect the DLDoS attack in real-time, which is constructed based on Extreme Learning Machine (ELM) [5], Whale Optimization Algorithm (WOA) [6], and the

Received 12 December 2024; revised 2 April 2025; accepted 8 July 2025; approved by IEEE Transactions on Networking Editor C. Peng. Date of publication 22 August 2025; date of current version 26 December 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62472153, in part by the Major Program of Xiangjiang Laboratory under Grant 22XJ01004, and in part by Hunan Provincial Natural Science Foundation of China under Grant 2025JJ50350. *(Corresponding author: Jiliang Zhang.)*

Dan Tang is with the College of Cyber Science and Technology, Hunan University (HNU), Changsha 410082, China, and also with the Xiangjiang Laboratory, China (e-mail: Dtang@hnu.edu.cn).

Xinmeng Li, Pei Tan, and Zheng Qin are with the College of Cyber Science and Technology, Hunan University (HNU), Changsha 410082, China (e-mail: lixinmeng@hnu.edu.cn; tanpei@hnu.edu.cn; zqin@hnu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York at New Paltz, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Jiliang Zhang is with the College of Integrated Circuit Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: zhangjiliang@njupt.edu.cn).

Digital Object Identifier 10.1109/TON.2025.3596117

idea of dual optimization. The Suspicious IP Location module determines suspicious IP based on double thresholds for attack mitigation. The Data Statistics and Suspicious IP Location module are deployed on the PDP using the Programming Protocol-Independent Packet Processors (P4) language [7].

In order to evaluate DOE-DTL, we carry out experiments in the simulation network built on Mininet [8] and Behavioral Model Version 2 (BMV2) [9] programmable switches. The results demonstrate that DOE-DTL has a high correct rate in the DLDoS attack detection, and can quickly discover and mitigate the attack (2s-6s), with low resource consumption.

In conclusion, the work has the following contributions:

- Put forward a system named DOE-DTL combining the PDP for the real-time DLDoS attack detection and mitigation in SDN.
- Offload part of the detection and mitigation work from the control plane to the PDP, making the pressure of the control plane less, the functions more flexible, and the time and resource cost lower.
- Train a DLDoS attack detection model used in the Attack Detection module with ELM and WOA based on a dual optimization strategy, the model judges according to the features extracted from the data counted on the PDP.
- Propose a method leveraging double thresholds to locate the suspicious IP on the PDP, which is useful for determining the attack sources in the DLDoS attack mitigation.
- Evaluate DOE-DTL in the simulation environment built on Mininet and BMV2 programmable switches and prove the effectiveness of the system.

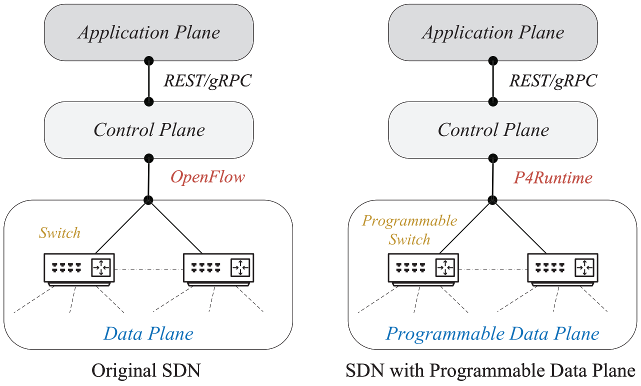
The following structure of this paper is: Section II discusses the background of our study. Section III introduces the related work. Section IV illustrates DOE-DTL in detail, including its architecture and its detection and mitigation strategies. Section V evaluates the performance of DOE-DTL. Section VI makes a conclusion of the work.

II. Background

A. Programmable Data Plane

Traditional network architectures employ rigidly integrated control plane and data planes, relying on fixed-function hardware devices for traffic management operations, so it is very inconvenient to redeploy network functions (requiring complex modifications). In order to deploy network functions more flexibly, SDN with decoupled control plane and data plane is proposed. The control plane has an open programming interface [10] to the data plane and programs the underlying hardware functions to allocate the network resources in the SDN.

In the first generation of SDN, the data plane itself is not programmable and its functions are deployed by the control plane through the communication based on the OpenFlow protocol [11]. Because the data plane can not expand the functions by itself, its flexibility is still low. With the development of technology, a new generation of SDN has been designed, which uses the PDP. The PDP enables autonomous protocol parsing and packet processing through its self-defined rulesets, eliminating control plane intervention. On the PDP, actions and the related parameters are maintained in the Match-Action Table, and the abstract logic is implemented through programming flow control programs [12]. This enhanced programmability has inspired novel approaches for attack detection and mitigation in SDN, prompting researchers to explore PDP-based solutions that alleviate control plane workload. Of course, in this generation of SDN, the control plane can still manage the functions of the PDP by deploying rules through the P4Runtime protocol [4], but the logic is different from the first generation of SDN. In the former generation, the devices of the data plane connect to the southbound interface open on the control plane, in contrast, the devices of the PDP open the gRPC server and then the control plane connects to them in the new generation. Fig. 1 shows the structure of the two generations of SDN.



Original SDN SDN with Programmable Data Plane

Fig. 1. Comparison of the two generations of SDN.

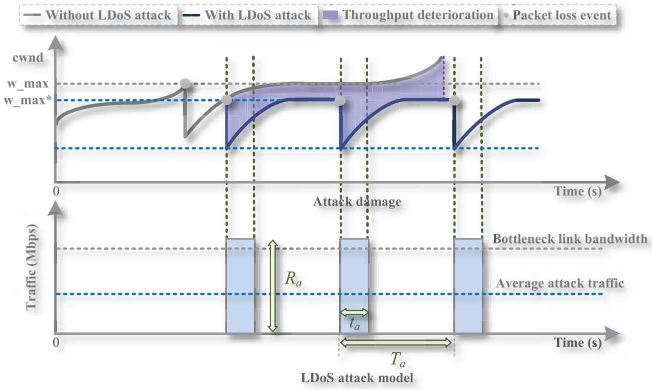


Fig. 2. Model and principle of the LDoS attack.

B. DLDoS Attack

In the LDoS attack, the attacker launches bursts with high speed and short duration in a periodic and discontinuous manner to preempt the link bandwidth, restricting the normal TCP traffic transmission. Specifically, the pulse traffic in LDoS attack leads to network congestion and packet loss, thereby triggering TCP congestion control mechanism. So that the available TCP transmission bandwidth continues to decrease as the congestion window (cwnd) is reduced. As a result, the normal TCP traffic cannot be transmitted smoothly. The attack

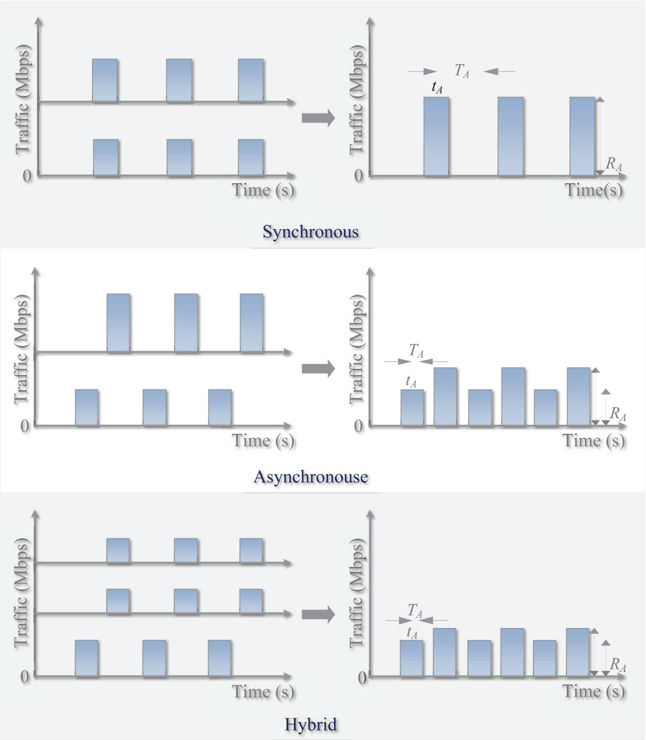


Fig. 3. Modes of the DLDoS attack.

traffic exhibits burst characteristics with peak rates during pulse periods, while maintaining an average volume below the bottleneck link bandwidth. Fig. 2 shows the model and principle of the LDoS attack. The LDoS attack is described using three key parameters [13]: attack cycle, pulse duration, and attack intensity, which are represented by T_a, t_a, and R_a respectively. And the attack damage is reflected by the change of the cwnd under the default congestion avoidance algorithm (CUBIC [14]) in the subsequent evaluation experiment environment (Linux system) of this paper.

The DLDoS attack is composed of LDoS attacks launched in a distributed manner, multiple attack sources launch LDoS attacks synchronously or asynchronously, the malicious attack flows aggregate in the bottleneck link to result in network congestion. Compared with traditional DDoS attacks, the DLDoS attack is more subtle with the lower average attack rate because of the short pulse time of each attack source. The DLDoS attack can be divided into three modes: synchronous, asynchronous, and hybrid, as shown in Fig. 3 (T_A, t_A, and R_A represent the attack cycle, pulse duration, and attack intensity of the aggregated attack traffic respectively). In the synchronous mode, multiple attack sources break out high-intensity pulses at the same time and attack the victim with the same T_a and t_a, so the total attack intensity R_A is higher. In the asynchronous mode, the attack sources break out attack pulses successively at different times, so the distribution of the attack pulses is denser. In the hybrid mode, the attack sources break out attack pulses synchronously or asynchronously, so R_A varies in the attack, and the attack pulses are also distributed relatively densely. Following the above division, we study the DLDoS attack detection and mitigation strategies. In the DLDoS attack, the network traffic distribution changes compared with the normal state, so we perform DLDoS attack detection and mitigation through traffic distribution change analysis in this paper.

III. Related Work

At present, some DLDoS attack detection and mitigation schemes have been proposed. Lei et al. carried out detection based on wavelet transform, the network traffic is decomposed and reconstructed for the DLDoS abnormal traffic identification [15]. Liu et al. created a detection algorithm utilizing data compression and behavior divergence, which measures the divergence degree of the network behavior based on Daub4 wavelet transform, and calculates the concentrated divergence energy percentage of each network traffic based on the weighted exponential moving average method to realize an accurate detection of DLDoS attack [16]. Sahoo et al. adopted the metrics based on the measurement method of generalized entropy to conduct the early detection of the DLDoS attack [17].

However, there is not enough study on the detection and mitigation of the DLDoS attack at present, and because DLDoS attack has a lower average rate and stronger concealment, the detection and mitigation methods applicable to traditional DDoS attacks are not effective against it. But some detection strategies such as those in [18], [19], and [20] can also be used for DLDoS attack detection. In addition, since each attack source launches an LDoS attack distributedly in the DLDoS attack, some detection and mitigation approaches of the LDoS attack are also suitable for detecting and mitigating the DLDoS attack after adjustment. Wu et al. detected the LDoS attack based on coherent detection, they calculate cross-correlation values and compare them with designed double threshold rules to determine whether the LDoS attack exists [21]. Yue et al. proposed that modeling the queue distribution to extract the attack feature and estimate the attack period on the basis of the queue behaviors for timely attack detection [22]. Xie et al. constructed a system named SoftGuard based on the ideas of threshold discrimination, cycle extraction, and sequence comparison to counter the LDoS attack in SDN [23]. Our team proposed a series of methods to respond to the LDoS attack, including some detection and mitigation systems for the LDoS attack in SDN, such as P&F [24] and HGB-FP [25].

For the proposed response methods in SDN, most of them are designed to be on the control plane completely. And to meet the needs of the detection and mitigation work, some necessary communication between the data plane and the control plane has to be carried out when using them, which brings unavoidable time and resource overhead. After the PDP is proposed, the attack detection and mitigation work can be directly deployed on it without the centralized management of the control plane, and the functions can be customized flexibly. According to that, there are some new studies. For example, Zhou et al. presented NetBeacon for the ML inference on the PDP, which can be used in the attack response [26]. da Silveira Ilha et al. designed a fine-grained and low-delay mechanism named Euclid based on the PDP to detect and mitigate DDoS attacks [27]. Laraba et al. resisted the hosts with improper behaviors by programming the switch rules to reduce the TCP protocol abuse and defend against network attacks [28]. Tang et al. designed a set of functional tools used on the PDP and proposed an in-network LDOS attack defense system [29]. Musumeci et al. implemented attack detection directly on the PDP based on the ML method [30]. Li et al. proposed a customizable system named POSEIDON to defend DDoS attacks, which modularizes defense strategies with defense primitives and deploys them on programmable switches to effectively defend against attacks [31]. Febro et al. deployed a new virtual network function on the edge programmable switches to protect network devices from the SIP DDoS attack [32]. Tavares and Ferreto proposed a SYN-flood attack defense system based on sketches, avoiding the shortcomings of the previous defense system, which would prolong client connection time and be vulnerable to buffer saturation attack [33]. Alcoz et al. proposed a congestion control mechanism named ACC-Turbo for Pulse-Wave DDoS attack. The mechanism continuously extracts data packet header characteristics at a linear rate to aggregate the network traffic, infer the probability that it is an attack flow cluster for each cluster, and formulate the packet scheduling policies on the PDP according to that [34]. However, as far as we know, almost all of the detection and mitigation methods utilizing the PDP at present are aimed at other types of network attacks, there is no scheme targets DLDoS attack. The system we design in this paper is specific to the DLDoS attack, and it conducts detection and mitigation in real-time combined with the PDP.

IV. System Design

A. Framework of DOE-DTL

Firstly, we want to systematically introduce the seven modules in DOE-DTL. Some of the modules are deployed on the control plane and some are on the PDP. Offloading some work to the PDP relieves the control plane to some extent, improves flexibility, saves some resources, and avoids partial delay.

1) Modules on the PDP: Data Statistics module. We analyze the traffic changes to determine the network status by obtaining the features that can reflect the distribution of network traffic. As described in Section II, the DLDoS attack can change the traffic distribution, the essence of which is that the number of packets and bytes of TCP and UDP traffic changes. Therefore, we obtain network traffic features based on the number of TCP packets, TCP bytes, UDP packets, and UDP bytes. The function of the Data Statistics module is to make real-time statistics of these numbers. Suspicious IP Location module. When determining that there is a DLDoS attack in the network, mitigation is conducted. In the mitigation, suspicious IP information is the indispensable basis, it is useful for identifying the attack source. This module locates suspicious IPs continuously from DOE-DTL starts based on double thresholds by programming the data packet processing logic, which is simple and fast. And it reports the suspicious IP information to the controller.

2) Modules on the Control Plane: Feature Extraction module. It polls the Data Statistics module to sample the statistical data and divides detection windows to obtain the network traffic features based on the sampled data. Traffic Monitoring module. To reduce the resource consumption of the system as far as possible, we do not directly input the acquired traffic features into the Attack Detection module for attack judgment. Instead, we design this module for pre-detection, and the calculation in it is simpler than the Attack Detection module. Introducing a pre-detection module may also shorten the time used to detect the DLDoS attack (less computation requires less computation time). Attack Detection module. The feature data is entered into it only when the Traffic Monitoring module has found that the DLDoS attack may exist. This module carries out fine-grained detection with a trained model based on the ML method according to the traffic features, determining whether the attack exists or not. Suspicious IP List Maintenance module. It processes the reported information from the Suspicious IP Location module on the PDP to obtain the suspicious IP. The suspicious IP is added to the suspicious IP list if it is not in, otherwise, it will be marked as the attack source IP if DOE-DTL is undergoing attack mitigation and the Mitigation Rule Deployment module will be informed to deploy appropriate rules according to this IP. Mitigation Rule Deployment module. It deploys rules based on the Match-Action Table stored on the PDP to filter the packets sent from the attack source IP and remove the mitigation rules from the table when the network state is back to normal.

Fig. 4 shows the framework of DOE-DTL. When DOE-DTL is running, on the PDP, the Data Statistics module counts the number of packets and bytes of TCP and UDP. The Suspicious IP Location module determines the suspicious IPs and reports the information to the Suspicious IP List Maintenance module. On the control plane, the Feature Extraction module calculates the network traffic features and chooses to input the data into the Traffic Monitoring module or the Attack Detection module according to a maintained label value fl (its initial value is 0). The Traffic Monitoring module is to carry out coarse-grained pre-detection (sending a signal to the Feature Extraction module when there may be an attack), while the Attack Detection module is to carry out fine-grained detection. The Suspicious IP List Maintenance module processes the reported information and maintains the suspicious IP list. When the Attack Detection module determines that the DLDoS attack exists in the network, it communicates with the Suspicious IP List Maintenance module and makes it start to determine the attack source IPs. The Mitigation Rule Deployment module deploys the mitigation rules according to the attack source IP information from the Suspicious IP List Maintenance module. When the Attack Detection module determines that the network has changed from an attack state to a normal state, it informs the suspicious IP List Maintenance module to reset and carry out a new round of maintenance work (avoiding the malicious interference caused by the historical information when locating the suspicious IP and determining the attack source IP), informs the Mitigation Rule Deployment module to remove the deployed mitigation rules from the Match-Action Table (avoid affecting the normal traffic forwarding), and informs the Feature Extraction module to reset fl to 0.

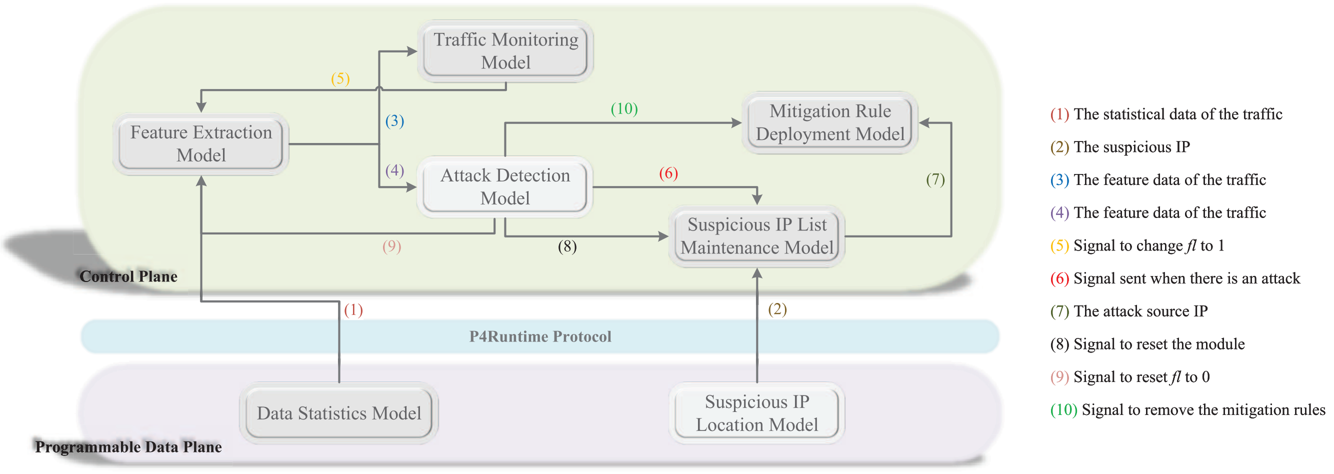

Traffic Monitoring Model
Feature Extraction Model
Mitigation Rule Deployment Model
Attack Detection Model
Suspicious IP List Maintenance Model
Control Plane
P4Runtime Protocol
Data Statistics Model
Suspicious IP Location Model
Programmable Data Plane
(1) The statistical data of the traffic
(2) The suspicious IP
(3) The feature data of the traffic
(4) The feature data of the traffic
(5) Signal to change fl to 1
(6) Signal sent when there is an attack
(7) The attack source IP
(8) Signal to reset the module
(9) Signal to reset fl to 0
(10) Signal to remove the mitigation rules

Fig. 4. The framework of DOE-DTL.

TABLE I
THE NOTATIONS IN THIS PAPER

Notation	Description
L	Nnumber of hidden layer neurons.
w_j	Connection weights of two layers.
b_j	Thresholds of the hidden layer.
x_i	Traffic feature data in training.
t_i	Label data in training.
$h_j(x_i)$	Output of hidden layer neuron.
H	Hidden-layer output matrix.
C	Regularization coefficient.
β^*	Optimal matrix of ELM.
M	Population number of WOA.
X^m, X^*, X^{m+}, X^r	Position vector of the individual.
f, f^m, f^*	Fitness value of each vector.
p, A, r_1, r_2, l	Random number in WOA.
t, T_{max}	The interation number.
a	Nonlinear convergence factor.
D	Median value in the WOA.
b	A constant in position caculation.
fl	Label stored in DOE-DTL.
s	Period of packet-count statistics.
Sm_{ip}, r_{ip}	Number of packets and times.
Smt	The packet-count threshold.
R, R_l	The number threshold.
Ct, Et_{ip}	Representation of time nodes.
$Flaga_{ip}$, $Flagb_{ip}$	Flags stored in registers.

Next, we will introduce the detection and mitigation strategies of DOE-DTL in detail, including the specific implementation methods of the functions of the above seven modules. The notations used in this paper are summarized in TABLE I.

B. Strategy in Attack Detection

1) Model Based on ELM: As described above, DOE-DTL uses a constructed model in the Attack Detection module to carry out DLDoS attack detection. For this model, we train it based on ELM and combine the idea of dual optimization.

ELM is a kind of ML model based on Single Hidden Layer Feedforward Neural Network (SLFN). While SLFN adjusts and determines the weights through the backpropagation algorithm in training, ELM employs the Moore-Penrose pseudoinverse to compute the hidden-to-output layer weight matrix, without iteration. In addition, its input-to-hidden layer weights and the hidden layer thresholds, are set in advance and no adjustment is required in training, which can also reduce part of the calculation amount, so the training process of ELM is relatively simple. On the other hand, the main work is a simple matrix operation when using the model based on ELM to detect, thus the calculation speed is fast and the attack can be found timely. Because of the simple training process and the fast calculation speed, we choose ELM as the basic model in the construction of the attack detection model.

When using the detection model based on ELM for DLDoS attack detection, the traffic features in the network are taken as the input layer, and ELM draws the result (the judgment of the network status) in the output layer through calculation. When training the detection model with the training data set, we take the traffic feature values $x_i|i = 1, 2, \ldots, n$ as the input layer and the corresponding labels (the attack exists or not) $t_i|i = 1, 2, \ldots, n$ as the output layer, and find the optimal hidden-to-output layer weight matrix β which minimizes the training error, denoted by β^*.

The following is the training process. First, multiply each input x_i by the weight $w_j|j = 1, 2, \ldots, L$ between the input layer neuron and the hidden layer neuron, plus the corresponding threshold b_j of the hidden layer (L is the neuron number in the hidden layer), and use the activation function *Sigmoid* to get the output $h_j(x_i)$ of the corresponding hidden layer neuron. The specific calculation formula is as formula (1). Then, calculate the hidden layer output matrix $H = [[h_1(x_1), \ldots, h_1(x_n)], \ldots, [h_L(x_1), \ldots, h_L(x_n)]]$, which is used to find β^*. We introduce the L_2 regularization term in the process of obtaining the optimal solution to avoid overfitting, and find β that makes $\frac{1}{2}||\beta||^2 + \frac{C}{2}||H\beta - T||^2$

minimum, where *C* is the regularization coefficient and *T* is the output matrix composed of label data t_i. According to the MP generalized inverse matrix theory, β^* can be calculated through the formula (2), where the solution result of $\left(H^T H + \frac{1}{C}\right)^{-1} H^T$ is the MP generalized inverse of *H*.

$$h_j(x_i) = \frac{1}{1 + e^{-(w_j x_i + b_i)}} \tag{1}$$

$$\beta^* = \left(H^T H + \tfrac{1}{C}\right)^{-1} H^T T \tag{2}$$

2) Dual Optimization of the Model: the values of w_j and b_j can affect the training effect in the model construction, so we adopt a dual optimization strategy to optimize the setting of these parameters. We use WOA, a new intelligent swarm optimization algorithm to search for the above parameters. The algorithm imitates the whale-preying behavior with few parameters and simple operations. At the same time, we adjust the factor convergence mode in WOA from linear convergence to nonlinear convergence and introduce adaptive weights in the position update, thereby improving the global search and local development ability for better search results.

In hunting, the whale that finds the prey first swims toward the prey, and its position is the initial optimal position. Other whales in the population constrictively encircle the prey or move in a spiral toward it with that whale. In the constrictive encirclement, each individual would choose to swim toward the whale in the optimal position, or toward a random whale. We represent the parameters that need to be preset in the form of vectors, regard them as the position vectors in WOA (the spatial dimension *d* is set to the total number of the parameters), and find the optimal vector according to the regulars of the position changes in the hunting of whale population. The following is the specific process.

a. Initialize the population number *M* and individual parameter vectors, the parameter vector of the individual $X^m | m = 1, 2, \ldots, M$ is $(X_1^m, X_2^m, \ldots, X_d^m)$. Calculate the fitness value f^m of each parameter vector (reflecting detection effect of the model trained by ELM under the setting of these parameters), and find the initial optimal parameter vector X^* (its fitness value *f* is optimal).

b. In each iteration, each individual parameter vector changes through the constrictive encirclement or the spiral position update according to the random number *p*. If $p < 0.5$, the former is chosen, to step c, otherwise the latter, to step e.

c. In the constrictive encirclement, the parameter vector determines to tend to whether the optimal vector or a random vector according to the random number *A* (the formula is as (3)), where r_1 and r_2 are the random values between 0 and 1, and *a* converges nonlinearly according to formula (4) as the number of iterations increases different from the traditional WOA whose *a* converges linearly from 2 to 0. t and T_{max} in the formula are the current iteration number and the maximum iteration number respectively. If $|A| \leq 1$, the parameter vector tends to the optimal vector, X^m updates to X^{m+} as formula (5), and the formula of *D* is (6). After the update, skip to Step f, and if $|A| > 1$, the parameter vector tends to a random vector, to step d.

$$A = 2ar_1 - a \tag{3}$$

$$a = \left(2 - \frac{2t}{T_{max}}\right)\left(1 - \frac{t^3}{T_{max}^3}\right) \tag{4}$$

$$X^{m+} = \left(\frac{t^3}{T_{max}^3}\right) X^m - AD \tag{5}$$

$$D = |2r_2 X^* - X^m| \tag{6}$$

d. Assume that it tends to the random vector X^r, the calculation formula for X^{m+} is as formula (7), and different from step c, the calculation formula for *D* is formula (8). After the position update is completed, skip to step f.

$$X^{m+} = \left(\frac{t^3}{T_{max}^3}\right) X^r - AD \tag{7}$$

$$D = |2r_2 X^r - X^m| \tag{8}$$

e. When the parameter vector changes through the spiral position update, it also tends to the optimal vector, but the formula is different as (9), where *b* is a constant, and *l* is a random number in the interval [−1, 1]. The solution formula of *D* is as (10). After the update, skip to Step f.

$$X^{m+} = De^{bl} cos(2\pi l) + \left(1 - \frac{t^3}{T_{max}^3}\right) X^* \tag{9}$$

$$D = |X^* - X^m| \tag{10}$$

f. Find the current X^*, and judge whether the maximum iteration number has been reached. If so, stop the search, and the current X^* is the result, otherwise, return to step b to continue the iteration.

The nonlinear decreasing convergence method of *a* makes WOA generate larger *A* in the early iteration stage to effectively improve the global exploration ability, and generate smaller *A* in the later stage to effectively improve the local development ability. In addition, the adaptive weight $\frac{t^3}{T_{max}^3}$ and $1 - \frac{t^3}{T_{max}^3}$ respectively in the constrictive encirclement and the spiral position update make the constrictive encirclement and the spiral position update can respectively have the smaller and the larger weights, or the larger and the smaller weights in the iterative optimization process, improving the global search ability and the local development ability.

For the calculation of *f*, we divide the training data set into two parts called TA and TB of similar size, ensuring that the proportion of the data under attack and not in the two parts is approximately the same. Then, we further divide TB into three parts called TB_1, TB_2, TB_3, and similarly, ensure that the three parts are similar in size as well as that the proportion of the data under attack and not is roughly the same. For each parameter vector in the iteration, we use TA to train the attack detection model based on ELM with these parameters, and use TB_1, TB_2, TB_3 to test the model to get *f*. We introduce $JaccardIndex(JI)$ [35], which can reflect the detection effect to calculate *f*. JI is calculated with the formula: $JI = DR/(DR + FPR + FNR)$, in which DR, FPR, and FNR are the detection rate, false positive rate, and false negative rate respectively (the detailed introduction is given

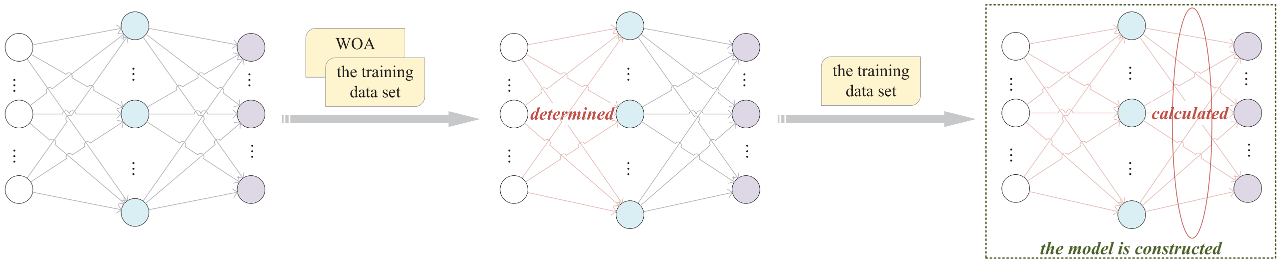


Fig. 5. The construction of the DLDoS attack detection model.

in Section V). f is calculated from the test results of TB_1, TB_2, and TB_3, as the formula: $f = (JI_1 + JI_2 + JI_3)/3$, where JI_1, JI_2, and JI_3 are the JI of TB_1, TB_2, and TB_3 respectively. In the process of parameter determination, we actually search for a parameter vector that makes f as large as possible, because a larger f means that the FPR and FNR of the model detection are lower, that is, the detection effect is better. Generally, we take ELM as the basic model to construct the DLDoS attack detection model and use WOA to pre-set its parameters (introduce a nonlinear convergence factor and adaptive weights). The flow of the DLDoS attack detection model construction is shown in Fig. 5.

3) Traffic Feature Extraction: the traffic features are needed for training and detection. For the feature extraction, in the Data Statistics module, we define two $packets\_and\_bytes\_Counter$, named $TCPCounter$ and $UDPCounter$. For the destination address (the object may be attacked), we count the number of packets and bytes of TCP and UDP sent to it. Specifically, for each packet, we extract its header information, determine its type (TCP or UDP), and count the corresponding *Counter*. In addition, to prevent data overflow, $TCPCounter$ and $UDPCounter$ are reset every 50s.

The Feature Extraction module polls $TCPCounter$ and $UDPCounter$, collects the number of packets and bytes of TCP and UDP traffic with the sampling window of 0.3s, and divides the detection windows in the form of the multiple-sliding window. Using the multi-sliding windows to make collaborative detection can improve the universality of DOE-DTL, so that it can well detect the DLDoS attack with different T_A and t_A (more timely and more accurate). We divided the detection windows with the step size 0.3s, and the length 1.5s, 3s, 4.5s, and 6s (called W_a, W_b, W_c and W_d respectively), and calculate the feature values of each window.

When the feature data is used in the training of DLDoS attack detection model, we attach a corresponding label to each group of data. If the sampling windows of attack in a detection window account for more than 80%, the feature data of this detection window is assigned label 1, indicating that it is the feature data of the traffic under DLDoS attack, otherwise, label 0 is assigned. Multiple groups of labeled feature data constitute the training data set in the model construction.

4) Attack Detection Process: When fl is 0, the Feature Extraction module sends the feature data into the Traffic Monitoring module, which makes an initial determination of the network status by comparing the feature data with a threshold (set according to the actual network traffic condition). When the Traffic Monitoring module determines that there may be an attack, it sends a signal to the Feature Extraction module to inform it to change fl to 1. When fl is 1, the Feature Extraction module sends the feature data to the Attack Detection module for the fine-grained detection using the attack detection model. For the four window division forms W_a, W_b, W_c, and W_d, if adjacent three detection windows of any form are determined to exist attack, the network is considered to be under the DLDoS attack, otherwise to be normal. The three-detection-window condition is set to further reduce the possibility of misjudgment. The attack detection algorithm is defined as Algorithm 1.

Algorithm 1 Attack Detection

```
Input: The label $fl$, the feature data $feature$, the threshold $trsod$, the statistical value $S_a$, $S_b$, $S_c$, $S_d$
Output: Detection result $R$
while $get(fl)$ do
    if $fl = 0$ then
        PreDetection($feature$, $trsod$);
    else
        AtkDetection($feature$, $trsod$);
Function PreDetection($feature$, $trsod$):
    Get the comparision $result$ of $feature$ and $trsod$;
    $fl \Leftarrow result$;
Function AtkDetection($feature$):
    Get the calculation result $r$ of the model;
    if $r = TRUE$ then
        if $feature$ $belong$ $to$ $W_a$ then
            $S_a \Leftarrow S_a + 1$;
        if $feature$ $belong$ $to$ $W_b$ then
            $S_b \Leftarrow S_b + 1$;
        if $feature$ $belong$ $to$ $W_c$ then
            $S_c \Leftarrow S_c + 1$;
        if $feature$ $belong$ $to$ $W_d$ then
            $S_d \Leftarrow S_d + 1$;
    if $S_a > 2$ $or$ $S_b > 2$ $or$ $S_c > 2$ $or$ $S_d > 2$ then
        $R \Leftarrow TRUE$;
    else
        $R \Leftarrow FALSE$;
    return $R$;
```

In DLDoS attack detection, the main work of the PDP is data statistics, which is lightweight. When the network is judged to be attacked, DOE-DTL conducts attack mitigation. In contrast to the attack detection, the PDP takes on the main work in the mitigation, which can avoid some communication overhead and accelerate the mitigation speed.

C. Strategy in Attack Mitigation

1) Suspicious Ip Location: The attack mitigation requires the joint participation of the Suspicious IP Location module, the Suspicious IP List Maintenance module, and the Mitigation Rule Deployment module. The Suspicious IP Location module locates suspicious IPs based on double thresholds according to the fact that the attack source sends a lot of UDP packets at the pulse time. The main idea is, for each very short period s (pulse-time level), it calculates the total UDP packets Sm_{ip} sent by each source IP. Continuously, the module counts the number of times that Sm_{ip} of each source IP reaches or exceeds the packet-count threshold *Smt*, which is represented by r_{ip}, and compares r_{ip} with the number threshold *R*. When r_{ip} is larger than or equal to *R*, the corresponding IP is judged as suspicious and reported, and to avoid reporting too much redundant information, a larger limit R_l is set, the IP is not reported when r_{ip} exceeds R_l. Reporting less redundant information can reduce the overhead of the reported information processing on the control plane. *R* is set to reduce the misjudgment. The subscript *ip* in the above notations is the index value corresponding to each source IP, which is calculated by hashing. *s*, *Smt*, *R*, and R_l are set and adjusted according to the actual network condition such as attack intensity to adapt to the dynamic network environment.

2) Attack Mitigation Process: When determining that a DLDoS attack exists in the network, the Attack Detection module sends a signal to the suspicious IP List Maintenance module. After receiving the signal, for each obtained suspicious IP, if it is already in the suspicious IP list and not labeled as an attack source IP, the suspicious IP List Maintenance module determines that it is an attack source IP and marks in the suspicious IP list, then, informs the Mitigation Rule Deployment module to deploy corresponding rules to drop the traffic issued by this IP refer to the Match-Action Table stored on the PDP. It is worth mentioning that the continuous operation of the Suspicious IP Location module makes it possible to find and report suspicious IP as soon as possible, so that when the DLDoS attack is determined to exist, DOE-DTL can quickly locate the attack source. And although the module is continuously running, its work is mainly based on the simple operations of *Register* such as read, write, and add, therefore, the resource consumption on the PDP is still at a low level, and the normal line-speed forwarding of data packets can be satisfied. When the Attack Detection module determines that the network changes from the attacked state to the normal state, it sends a signal to the suspicious IP List Maintenance module, Mitigation Rule Deployment module, and the Feature Extraction module respectively. After receiving the signal, the suspicious IP List Maintenance module clears the suspicious IP list and starts a new round of maintenance work, the Feature Extraction module resets *fl* to 0.

Algorithm 2 Suspicious IP Determination

```
Input: $Register_{packets}$, $Register_{num}$, $Register_{time}$, $s$,
       $Smt$, $R_l$, $R$, the source IP of the packet $ip$, the
       current time $Ct$
Output: Suspicious IP $ip$
Initialize $index$, $Et$, $fa$, $fb$, $Sm$, $r$;
$index \leftarrow hash(ip)$;
$Et \leftarrow Register_{time}[index]$;
$Sm \leftarrow Register_{packets}[index]$;
$Sm \leftarrow Sm + 1$;
if $Ct - Et < s$ then
    $fa \leftarrow 0$ ;
    $Register_{time}[index] \leftarrow Sm$;
else
    $fa \leftarrow 1$ ;
    $Register_{packets}[index] \leftarrow 0$;
    $Register_{time}[index] \leftarrow Ct$;
    if $Sm < Smt$ then
        $fb \leftarrow 0$ ;
    else
        $fb \leftarrow 1$ ;
if $fb = 1$ then
    $r \leftarrow Register_{num}[index]$;
    if $r <= R_l$ then
        $r \leftarrow r + 1$;
        $Register_{num}[index] \leftarrow r$;
        if $r >= R$ then
            return $ip$;
```

In order to satisfy the need for read and write operations in the calculation, we choose to use the *Register* (the *Counter* mentioned before is no longer applicable). We define 5 registers, $Register_{packets}$, $Register_{num}$, $Register_{time}$, $Register_{flaga}$, and $Register_{flagb}$ to record the Sm_{ip} and r_{ip} in each s, the end time Et_{ip} of last s and the flags in the judgment of each source IP separately. In addition, for s, we use the *timestamp* to divide. The specific implementation of the function of the Suspicious IP Location module is as follows. For each arrival data packet, we get the current time *Ct* from the *timestamp* metadata it carries and identify whether it is a UDP packet by extracting its packet header information. If so, we hash the source IP to get the index value *ip*, and then, read the Et_{ip} at *ip* position of $Register_{time}$ and determine whether the interval of the value and *Ct* reaches or exceeds s. If not reach s, we set the $Flaga_{ip}$ at *ip* position of $Register_{flaga}$ to 0, otherwise, we set it to 1 and update the *Et* at the corresponding position of $Register_{time}$ to *Ct*. Each time a UDP packet arrives, we read the Sm_{ip} at *ip* position of $Register_{packets}$ and add 1 to it. And when $Flaga_{ip}$ is 0, we rewrite the new Sm_{ip}, when $Flaga_{ip}$ is 1, we compare Sm_{ip} with *Smt* to get the $Flagb_{ip}$ (0 for less than *Smt*, 1 for larger than or equal to *Smt*) and reset *ip* position of $Register_{packets}$ to 0. $Flagb_{ip}$ is stored at *ip* position of $Register_{flagb}$. If $Flagb_{ip}$ is 1, the r_{ip} at the corresponding position of $Register_{num}$ is read. When r_{ip} is larger than R_l, no operation is carried out, otherwise, we add 1 to r_{ip} to obtain a new value and write it to the original location. In addition, if the new r_{ip} is larger than or equal to *R*, we judge the corresponding source IP to be suspicious and use *Digest* to report it to the control plane. The algorithm to determine when a UDP packet arrives is described as Algorithm 2.

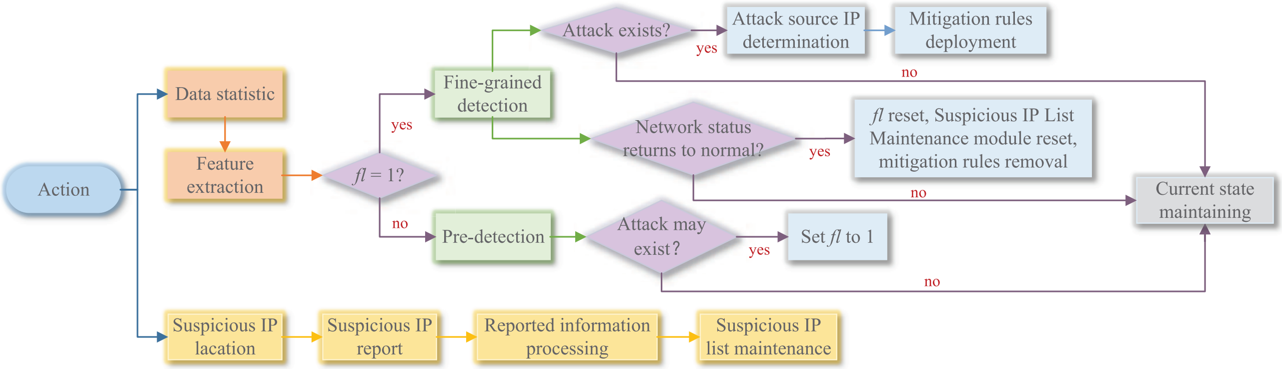

Action
Data statistic
Feature extraction
fl = 1?
yes
no
Fine-grained detection
Attack exists?
yes
no
Attack source IP determination
Mitigation rules deployment
Network status returns to normal?
yes
no
fl reset, Suspicious IP List Maintenance module reset, mitigation rules removal
Current state maintaining
Pre-detection
Attack may exist?
yes
no
Set *fl* to 1
Suspicious IP lacation
Suspicious IP report
Reported information processing
Suspicious IP list maintenance

Fig. 6. The detection and mitigation flow of DOE-DTL.

In fact, the PDP and the control plane communicate only when there is suspicious IP information to report or mitigating rules to deploy, producing a low time overhead. Without a doubt, attack detection is a necessary prerequisite for mitigation, so we make a summary of the overall work process of DOE-DTL in Fig. 6, including detection and mitigation.

D. Complexity Analysis

We analyze the computational complexity of DOE-DTL, including time complexity (TC) and space complexity (SC). The main calculation work is in the Data Statistics module, the Feature Extraction module, the Attack Detection module, and the Suspicious IP Location module.

In the Data Statistics module, the number of packets and bytes of TCP and UDP are counted. Each statistical operation needs to extract and parse the packet header of the arrived packet, and then carry out the corresponding counting operation, the statistical information is stored in $TCPCounter$ and $UDPCounter$. The TC and SC are $O(1)$ and $O(n)$ respectively. In the Feature Extraction module, the data in detection windows is sampled by polling $TCPCounter$ and $UDPCounter$, with subsequent extraction of the feature values for each window. The time overhead is mainly generated by the feature calculation, the TC is $O(n)$. And the space overhead is mainly generated by the storage of the sampled data, the SC is $O(n)$. In the Attack Detection module, the feature data is input to the attack detection model, and the determination results are obtained through the calculation in the model. The trained attack detection model is essentially a single-hidden-layer neural network with fixed weights and thresholds. The TC of the computation is $O(n)$, and the SC is $O(n^2)$. In the Suspicious IP Location module, when a UDP packet arrives, a count operation is performed, and when the interval time reaches s, a comparison operation is performed, the TC is $O(1)$. The $Registers$ are used for counting, so the SC is $O(n)$.

On the whole, the TC and SC of DOE-DTL are $O(n)$ and $O(n^2)$ respectively. Therefore, the computational complexity of DOE-DTL is low, confirming that DOE-DTL achieves an effective balance between accuracy and computational efficiency.

V. Experiments and Analysis

A. Experimental Setup

We deployed DOE-DTL in a simulation network based on Mininet and BMV2 programmable switches and conducted a series of performance evaluation experiments. The operating system is Linux Ubuntu 20.14, the virtual machine is VMware 16.0.0, and the memory RAM and the hard disk are 8GB and 64GB respectively. In the simulation network, the controller implements functions by running Python scripts and communicates with the programmable switches based on the P4Runtime protocol, and the network topology is as Fig. 7. We use the real network Topology Sprint from Internet Topology Zoo [36] as the evaluation network topology, which consists of 11 cities and 18 city links, each city is treated as a switch, the bandwidth of the city links is 45Mbps, and the delay of each city link is marked in Fig. 7, in addition, each city connects to local users, the bandwidth of the local links is 1Gbps, and the delay of each local link is 0ms.

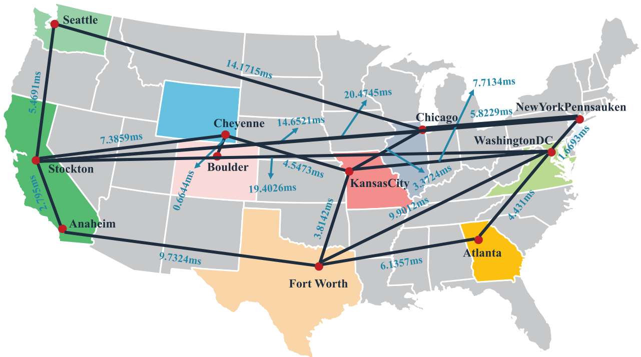
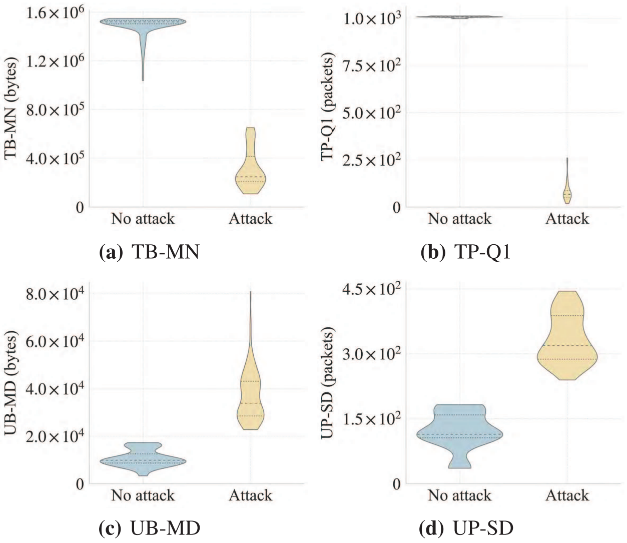


Fig. 7. Network topology for evaluation experiments in SDN.

It is stated here that in order to avoid unnecessary financial consumption, we don't configure a hardware programmable switch for the experiments. Because the hardware and software are functionally equivalent [27], we believe that the evaluation results in the simulation environment are sufficient to truly reflect the performance of DOE-DTL. In the experiments, we use Tcpreplay [37] to replay MAWI dataset [38] as background traffic, and run Python sockets to generate attack traffic. MAWI dataset is captured from the WIDE Project.

For the features to be acquired by the Feature Extraction module, we choose that can reflect the network traffic changes in the experiment. Meanwhile, in order to consume less feature calculation time, we choose the features as simple as possible. In the DLDoS attack, each attacker sends UDP streams at pulse time to attack, and under the action of the congestion control mechanism, the normal transmission of TCP traffic is affected. Therefore, the dispersion degree of UDP packets



(a) TB-MN

(b) TP-Q1

(c) UB-MD

(d) UP-SD

Fig. 8. The feature values under DLDoS attack and not.

TABLE II
THE ATTACK PARAMETERS OF THE EXPERIMENTS

Group Number	T_a(s)	t_a(s)	R_a(Mbps)
1	1.0	0.225	55
2	1.0	0.250	50
3	1.0	0.245	55
4	1.0	0.225	60
5	1.5	0.350	55
6	1.5	0.325	55
7	1.5	0.400	50
8	1.5	0.300	60
9	2.0	0.475	55
10	2.0	0.450	55
11	2.0	0.525	50
12	2.0	0.475	55

and bytes becomes large, and the number of TCP packets and bytes decreases significantly in the DLDoS attack. We select $TB-MN$, $TP-Q1$, $UB-MD$, and $UP-SD$ as the features, which are respectively the mean TCP bytes, the first quartile of TCP packets, the mean deviation of UDP bytes and the standard deviation of UDP packets. The feature values under DLDoS attack and not are shown in Fig. 8. In the Traffic Monitoring module, we use $TB-MN$ as the pre-detection feature. When the value of $TB-MN$ is less than 1000000, it is determined that there may be a DLDoS attack and the Attack Detection module is started to be used.

The attack parameters in the experiment are summarized in TABLE II, each attack source selects one group in it as the attack parameters when attacks. We use these parameters to launch DLDoS attacks, collect network traffic data in both normal and attack states and process them to obtain the training set for the attack detection model. And we set s and *Smt* in the suspicious IP location as 0.2s and 10000 according to these parameters. Meanwhile, in order to evaluate the fastest mitigation speed that DOE-DTL can reach, we set R to 1, that is, the source IP will be decided to be a suspicious IP as long as the number of UDP packets sent by it exceeds *Smt* in a certain s, and for R_l, we set it to 10.

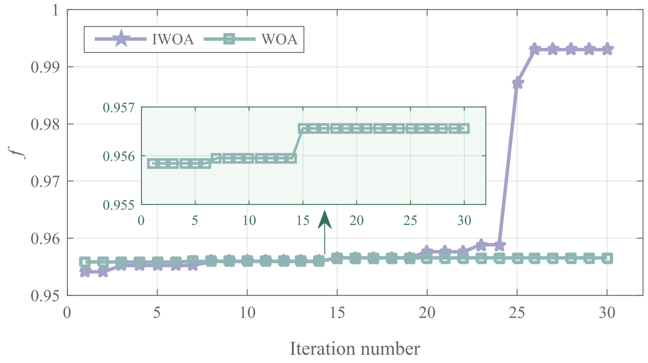


Fig. 9. IWOA vs. WOA.

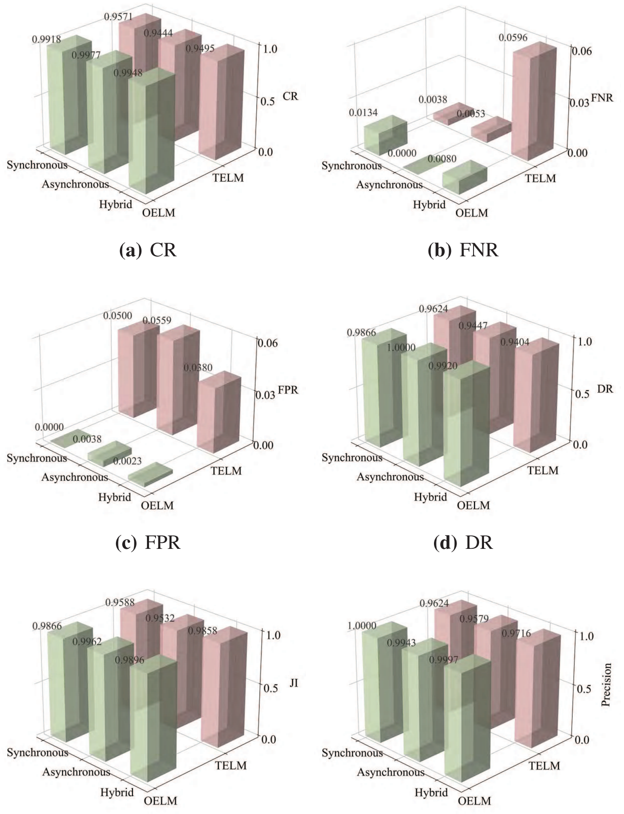
B. Evaluation Metrics

We select a series of evaluation metrics in the performance evaluation of DOE-DTL. In the evaluation of the detection effect, in addition to DR, FPR, FNR, and JI mentioned in Section IV, we also use the correct rate CR and *Precision*. We use these metrics to consider both the cases of the false negative and the false positive for more comprehensive evaluations of the detection effect. The lower FPR and FNR are, and the higher the other metrics are, the better the detection effect is. In terms of response time evaluation, we set attack detection time DT and attack mitigation time MT as evaluation metrics. In addition, we also set a metric ST to evaluate the overall time consumption of attack detection and mitigation. In terms of resource occupation evaluation, CU and MU are selected to evaluate the CPU and memory usage of DOE-DTL during operation. The calculation formulas of CR, FNR, FPR, DR, and *Precision* are as follows: $CR = (TN + TP)/(TN + TP + FN + FP)$, $FNR = FN/(TP + FN)$, $FPR = FP/(TN + FP)$, $DR = TP/(TP + FN)$, and $Precision = TP/(TP + FP)$. TP, TN, FP, and FN respectively represent the number of times that attacks are correctly determined to exist, the number of times that attacks are correctly determined to not exist, the number of false positives and the number of false negatives.

C. Performance Evaluation and Comparison

1) Optimization Effect of WOA: As described in Section IV, we optimize the basic WOA to improve its global search capability and local exploitation capability. We perform experiments to verify the effect of the optimization. Fig. 9 shows the change of f when selecting parameters based on traditional WOA (WOA) and improved WOA (IWOA). It can be seen that IWOA can achieve a higher f, that is, better parameters can be searched.

2) Effectiveness of Dual Optimization: In the training of the attack detection model used by DOE-DTL, we adopt the idea of dual optimization. In order to judge whether the dual optimization strategy makes the constructed detection model more effective, we also construct an attack detection model based on the basic ELM. Here, we respectively use OELM and TELM to indicate the attack detection model constructed with parameter setting and not. By comparing the detection effect of the DLDoS attack of OELM and TELM, we verify the significance of dual optimization. Fig. 10 shows the comparison between the two. On the whole, CR, DR, JI,



(a) CR **(b)** FNR

(c) FPR **(d)** DR

(e) JI **(f)** Precision

Fig. 10. OELM vs. TELM.

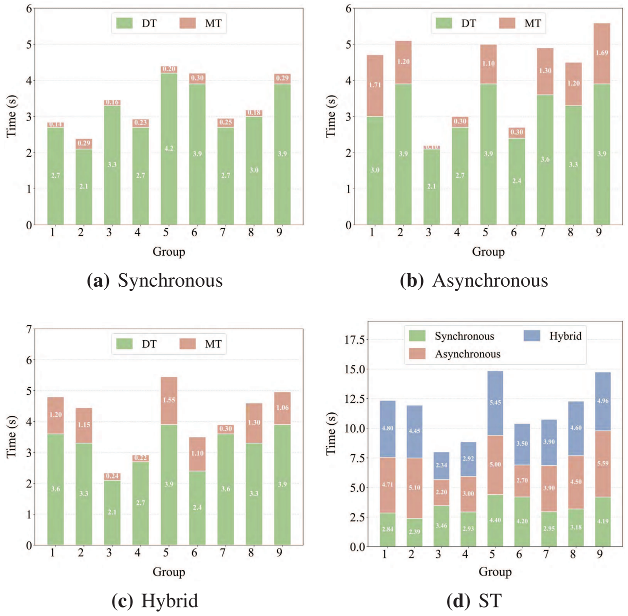
TABLE III
THE RESULTS OF DETECTION EFFECT EVALUATION

	Synchronous	Asynchronous	Hybrid	Avg.
CR	0.9918	0.9977	0.9948	0.9948
FNR	0.0134	0.0	0.0080	0.0071
FPR	0.0	0.0038	0.0023	0.0020
DR	0.9866	1.0	0.9920	0.9929
JI	0.9866	0.9962	0.9896	0.9908
Precision	1.0	0.9943	0.9977	0.9973

and *Precision* are higher in OELM, while *FNR* and *FPR* are lower.

3) Detection Effect: The results in TABLE III show that for all three attack modes, DOE-DTL has a good detection effect. On average, *CR*, *DR*, *Precision*, *JI* can reach more than 0.99, and *FNR* and *FPR* are less than 0.01, indicating that DOE-DTL has a very low probability of the false positive and false negative. DOE-DTL can not only detect the DLDoS attack in time but also avoid misjudgment of benign traffic as much as possible.

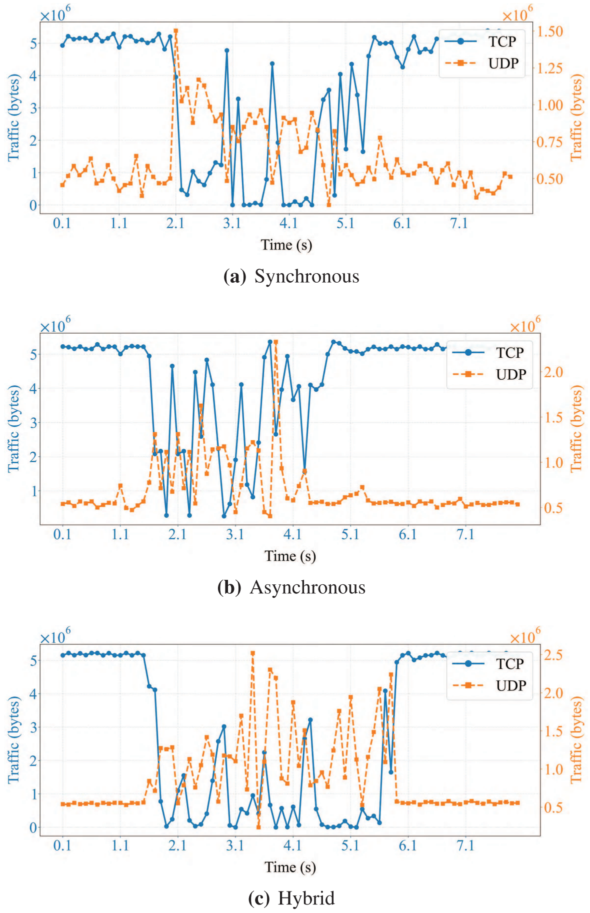
4) Response Time: For synchronous, asynchronous, and hybrid DLDoS attack, we respectively conduct 9 groups of experiments to evaluate the detection time and mitigation time spent by DOE-DTL in the work. Fig. 11(a), 11(b), and 11(c) respectively show the *DT* and *MT* of the system against synchronous, asynchronous, and hybrid attack. And Fig. 11(d) shows the *ST* of DOE-DTL under the three attack modes.



(a) Synchronous **(b)** Asynchronous

(c) Hybrid **(d)** ST

Fig. 11. The results of response time evaluation.



(a) Synchronous

(b) Asynchronous

(c) Hybrid

Fig. 12. The traffic changes in mitigation.

Generally speaking, DOE-DTL takes about 2s-6s in total to detect and mitigate the DLDoS attack, which is pretty quick.

We capture the traffic in the bottleneck link and draw images to observe the traffic changes under the DLDoS attack when DOE-DTL is deployed. Fig. 12(a), Fig. 12(b), and Fig. 12(c) show the changes when the attack is synchronous, asynchronous, and hybrid respectively. After the attack starts,

TABLE IV
COMPARISON WITH OTHER METHODS

System	ST(s)	CU(%)	MU(MB)
SoftGuard [23]	12-24	2-4	28-36
P&F [24]	6-22	2-4	24-30
HGB-FP [25]	2-33	2-12	–
NetBeacon [26]	3-7	–	–
PLUTO [29]	3-4	–	–
DOE-DTL	**2-6**	**0.5-3.8**	**30**

the transmission of normal TCP traffic is affected and its proportion in the bottleneck link is reduced. Subsequently, through the detection and mitigation mechanisms of DOE-DTL, the attack phenomenon is discovered, and the attack source IPs are pinpointed, then the traffic from the attack source IPs is filtered. The network traffic transmission finally returns to the normal level. In the period when the network traffic distribution is abnormal, DOE-DTL detects the attack, locates attack source IPs, and deploys mitigation rules to drop attack packets. We can see that the traffic can recover to normal soon under the detection and mitigation of DOE-DTL.

5) Resource Occupation and Comparison With Other Systems: As shown in TABLE IV, in the process of detection and mitigation, CU of DOE-DTL is between 0.5% and 3.8%, and MU is around 30MB. Both CU and MU are lower, which indicates that DOE-DTL is a relatively lightweight system for detecting and mitigating the DLDoS attack in SDN.

As far as we are aware, no prior research has investigated DLDoS attack detection and mitigation strategies in SDN combined with PDP. So we compare DOE-DTL with some similar detection and mitigation systems in SDN, they are SoftGuard [23], P&F [24], and HGB-FP [25]. These systems are for the LDoS attack and little work of them is deployed on the data plane. Except for CU and MU, TABLE IV shows that the ST of DOE-DTL is short, although its lower-bound is a little longer than HGB-FP, its upper-bound is far shorter than all the comparison objects, that is, it remains at a low level, which is more valuable. Besides, we compare DOE-DTL with the relevant methods on the PDP, NetBeacon [26] and PLUTO [29], and DOE-DTL is on par with them in terms of ST. The advantages in time cost of the combination with the PDP in the system design is reflected.

VI. CONCLUSION AND FUTURE WORK

In this paper, we design a system named DOE-DTL for the detection and mitigation of the DLDoS attack in SDN, combined with the PDP. In DOE-DTL, an attack detection model based on ELM and a dual optimization strategy utilizing WOA is used to detect the DLDoS attack in real-time. And DOE-DTL locates the suspicious IP based on double thresholds to further identify attack sources and deploy mitigation rules. In the simulation environment built on Mininet and BMV2 programmable switches, we evaluated DOE-DTL from three aspects: detection effect, response time, and resource occupation. The results show that DOE-DTL is an effective system for the real-time DLDoS attack detection and mitigation, and because it offloads part of the work to the PDP, the functions in it are flexible and the time and resource consumption in the work is low.

In the future work, we will strive to move more work to the PDP to further leverage the benefits of deploying work on the data plane, and adapt DOE-DTL to be suitable for detecting and mitigating more types of attacks. Additionally, the modular design idea of DOE-DTL facilitates the adjustment of each module, we will choose more basic ML models to build the attack detection model in the Attack Detection module and analyze the corresponding impact on the system performance.

REFERENCES

[1] J. Kim et al., "Enhancing security in SDN: Systematizing attacks and defenses from a penetration perspective," *Comput. Netw.*, vol. 241, Mar. 2024, Art. no. 110203.

[2] D. Tang, R. Dai, Y. Yan, K. Li, W. Liang, and Z. Qin, "When SDN meets low-rate threats: A survey of attacks and countermeasures in programmable networks," *ACM Comput. Surveys*, vol. 57, no. 4, pp. 1–32, Apr. 2025.

[3] V. Hnamte, A. A. Najar, H. Nhung-Nguyen, J. Hussain, and M. N. Sugali, "DDoS attack detection and mitigation using deep neural network in SDN environment," *Comput. Secur.*, vol. 138, Mar. 2024, Art. no. 103661.

[4] A. Liatifis, P. Sarigiannidis, V. Argyriou, and T. Lagkas, "Advancing SDN from OpenFlow to p4: A survey," *ACM Comput. Surveys*, vol. 55, no. 9, pp. 1–37, Sep. 2023.

[5] S. Ding, X. Xu, and R. Nie, "Extreme learning machine and its applications," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 549–556, Sep. 2014.

[6] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[7] Y.-C. Wang and P.-Y. Su, "Collaborative defense against hybrid network attacks by SDN controllers and P4 switches," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 2, pp. 1480–1495, Mar. 2024.

[8] B. Lantz. (2009). *Mininet*. [Online]. Available: http://mininet.org/

[9] P. L. Consortium. (2018). *Behavioral Model (BMv2)*. [Online]. Available: https://github.com/p4lang/behavioral-model

[10] D. Tang, R. Dai, C. Zuo, J. Chen, K. Li, and Z. Qin, "A low-rate DoS attack mitigation scheme based on port and traffic state in SDN," *IEEE Trans. Comput.*, vol. 74, no. 5, pp. 1758–1770, May 2025.

[11] C. Wang, "Dos attack mitigation in openflow and p4 programmable data planes," M.S. thesis, Dept. Comput. Sci., North Carolina Agricultural and Technical State University, Greensboro, NC, USA, 2023.

[12] Z. Xu, Z. Lu, and Z. Zhu, "Information-sensitive in-band network telemetry in P4-based programmable data plane," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 5081–5096, Dec. 2024.

[13] D. Tang, S. Wang, B. Liu, W. Jin, and J. Zhang, "GASF-IPP: Detection and mitigation of LDoS attack in SDN," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3373–3384, Sep. 2023.

[14] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008.

[15] G. Lei, L. Ji, R. Ji, Y. Cao, W. Yang, and H. Wang, "Can wavelet transform detect LDDoS abnormal traffic in multipath TCP transmission system?," *Secur. Commun. Netw.*, vol. 2021, pp. 1–8, Dec. 2021.

[16] X. Liu, J. Ren, H. He, Q. Wang, and C. Song, "Low-rate DDoS attacks detection method using data compression and behavior divergence measurement," *Comput. Secur.*, vol. 100, Jan. 2021, Art. no. 102107.

[17] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. P. C. Rodrigues, B. Sahoo, and R. Dash, "An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics," *Future Gener. Comput. Syst.*, vol. 89, pp. 685–697, Dec. 2018.

[18] W. Liang, L. Xiao, K. Zhang, M. Tang, D. He, and K.-C. Li, "Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14741–14751, Aug. 2022.

[19] X. Li et al., "Online Internet anomaly detection with high accuracy: A fast tensor factorization solution," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1900–1908.

[20] D. Tang, Y. Yan, C. Gao, W. Liang, and W. Jin, "LtRFT: Mitigate the low-rate data plane DDoS attack with learning-to-rank enabled flow tables," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3143–3157, 2023.

[21] Z. Wu, Y. Yin, G. Li, and M. Yue, "Coherent detection of synchronous low-rate DoS attacks," *Secur. Commun. Netw.*, vol. 2021, pp. 1–14, Mar. 2021.

[22] M. Yue, Z. Wu, and J. Wang, "Detecting LDoS attack bursts based on queue distribution," *IET Inf. Secur.*, vol. 13, no. 3, pp. 285–292, May 2019.

[23] R. Xie, M. Xu, J. Cao, and Q. Li, "SoftGuard: Defend against the low-rate TCP attack in SDN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[24] D. Tang, Y. Yan, S. Zhang, J. Chen, and Z. Qin, "Performance and features: Mitigating the low-rate TCP-targeted DoS attack via SDN," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 428–444, Jan. 2022.

[25] D. Tang, S. Zhang, Y. Yan, J. Chen, and Z. Qin, "Real-time detection and mitigation of LDoS attacks in the SDN using the HGB-FP algorithm," *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3471–3484, Nov. 2022.

[26] G. Zhou, Z. Liu, C. Fu, Q. Li, and K. Xu, "An efficient design of intelligent network data plane," in *Proc. 32nd USENIX Secur. Symp.*, 2023, pp. 6203–6220.

[27] A. D. S. Ilha, Â. C. Lapolli, J. A. Marques, and L. P. Gaspary, "Euclid: A fully in-network, P4-based approach for real-time DDoS attack detection and mitigation," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 3, pp. 3121–3139, Sep. 2021.

[28] A. Laraba, J. François, S. R. Chowdhury, I. Chrisment, and R. Boutaba, "Mitigating TCP protocol misuse with programmable data planes," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 760–774, Mar. 2021.

[29] D. Tang, B. Liu, K. Li, S. Xiao, W. Liang, and J. Zhang, "PLUTO: A robust LDoS attack defense system executing at line speed," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 3, pp. 2855–2872, May 2025.

[30] F. Musumeci, V. Ionata, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-learning-assisted DDoS attack detection with P4 language," in *Proc. ICC - IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[31] G. Li et al., "Enabling performant, flexible and cost-efficient DDoS defense with programmable switches," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1509–1526, Aug. 2021.

[32] A. Febro, H. Xiao, J. Spring, and B. Christianson, "Edge security for SIP-enabled IoT devices with P4," *Comput. Netw.*, vol. 203, Feb. 2022, Art. no. 108698.

[33] K. Tavares and T. Ferreto, "DDoS on sketch: Spoofed DDoS attack defense with programmable data plans using sketches in SDN," in *Proc. Anais do 37th Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, May 2019, pp. 805–819.

[34] A. G. Alcoz, M. Strohmeier, V. Lenders, and L. Vanbever, "Aggregate-based congestion control for pulse-wave DDoS defense," in *Proc. ACM SIGCOMM Conf.*, Aug. 2022, pp. 693–706.

[35] J. Liang and M. Ma, "ECF-MRS: An efficient and collaborative framework with Markov-based reputation scheme for IDSs in vehicular networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 278–290, 2021.

[36] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011. [Online]. Available: http://www.topology-zoo.org/dataset.html

[37] *Tcpreplay*. Accessed: Jan. 1980. [Online]. Available: https://tcpreplay.appneta.com/

[38] (2020). *Mawi Public Dataset*. [Online]. Available: http://mawi.wide.ad.jp/mawi/samplepoint-G/2020/

Dan Tang received the Ph.D. degree from the Huazhong University of Science and Technology in 2014. He is an Associate Professor at the College of Cyber Science and Technology, Hunan University (HNU), Changsha, China. His research interests include computer network security, computer information security, and architecture of future internet.

Xinmeng Li received the B.E. degree in information security and the master's degree in computer science and technology from Hunan University in 2022 and 2025, respectively. Her research interests include SDN, programmable data plane, network security, and network attack response.

Pei Tan received the B.E. degree in computer science and technology from Hunan University in 2024, where she is currently pursuing the master's degree with the College of Cyber Science and Technology. Her research directions include programmable networks and network security.

Keqin Li (Fellow, IEEE) is a SUNY Distinguished Professor of computer science with the State University of New York at New Paltz and also a National Distinguished Professor with Hunan University. His current research interests include cloud computing, high-performance computing, computer networking, and machine learning. He is an AAAS Fellow and an AIIA Fellow. He is a member of European Academy of Sciences and Arts. He is a member of Academia Europaea (Academician of the Academy of Europe). He was listed in ScholarGPS Highly Ranked Scholars (2022–2024) and is among the top 0.002% out of more than 30 million scholars worldwide based on a composite score of three ranking metrics for research productivity, impact, and quality in the recent five years.

Zheng Qin (Associate Member, IEEE) received the Ph.D. degree in computer software and theory from Chongqing University, China, in 2001. He is currently a Professor of computer science and technology at Hunan University, China. He has accumulated rich experience in product development and application services, such as financial, medical, and military. His main interests include computer networks and information security, cloud computing, and software engineering. He is a member of China Computer Federation (CCF) and ACM.

Jiliang Zhang (Senior Member, IEEE) joined the Integrated Circuit Science and Engineering College, Nanjing University of Posts and Telecommunications, currently serving as the College's President in 2025, and is also the Chair of the CCF Fault-Tolerant Computing Professional Committee. He works on power/energy efficiency and security problems in the design of integrated circuits (IC), processors, and the Internet of Things. He has authored more than 100 technical articles in leading journals and conferences. He has been the Program Committee Member for a number of well-known conferences, such as DAC, ASP-DAC, GLSVLSI, and FPT. He was a recipient of the CCF Integrated Circuit Early Career Award and the Winner of the Excellent Youth Fund of the National Natural Science Foundation of China. He served as an Associate Editor for some scientific journals, including IEEE Transactions on Circuits and Systems—I: Regular Papers, IEEE Transactions on Circuits and Systems—II: Express Briefs, and JEIT.