The Evaluation of DDoS Attack Effect Based on Neural Network

Wei Guo, Han Qiu, Zimian Liu, Junhu Zhu, and Qingxian Wang

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China

Correspondence should be addressed to Han Qiu; qiuhan_loach@sina.com

Received 2 March 2022; Accepted 22 March 2022; Published 11 April 2022

Copyright © 2022 Wei Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

DDoS attack effect evaluation is the basis of security strategy deployment. The traditional effect evaluation method relies on the original data, ignoring the relationship between features and the evaluation target and indicator data redundancy, which affects the accuracy and reliability of the evaluation result. To this end, we introduce distance entropy to measure the similarity between features and evaluation target and use LSTM and Triplet networks to measure multiple correlations simultaneously. Then, a 2D-CNN is used to mine deep feature information and filter irrelevant information. We also combine 1D-CNN and attention models to achieve hierarchical sampling of different local features. Finally, three fully connected layers’ training obtains a total evaluation value. We conducted experiments on five commonly used DDoS datasets. The results showed that the average ranking accuracy of the neural network-based DDoS attack evaluation method (NNDE) reached 87.2%, 91.3%, 88%, 85.6%, and 94.5%, respectively. Compared with other evaluation methods, an average increase of 19.73% indicates that this method can better evaluate the effect of DDoS attacks.

1. Introduction

The Distributed Denial of Service Attack (DDoS) has become one of the most threatening and lethal network attacks [1]. Its attack targets are usually critical Internet infrastructure or computing devices that provide essential services to outsiders. These devices include communication devices such as routers and gateways and high-performance servers that provide services such as education and banking. There are two characteristics among these intermediate nodes or terminals. One is to provide uninterrupted hardware support for diversified network operations and user access. The second is to verify the source of data packets by checking simple information such as source IP. Due to the lack of necessary traffic filtering methods, hackers can easily pretend to be ordinary users through IP spoofing [2]. They can also use reflection attacks to continuously send large-flow data packets to the target end to block normal services [3].

The low threshold and high yield of DDoS attacks make it possible for a fledgling novice to become an aggressive hacker overnight. Furthermore, their ignorance and aimlessness increase the frequency of cyberattacks. In order to cope with this network attack that may occur abruptly, the current intrusion detection system must constantly maintain a high alert state [4]. This comprehensive security strategy lacks threat grading. It treats all situations and periods as equally important, wasting a lot of extra labor and materials. Consequently, it is needed to evaluate the effect of DDoS attacks during an early attack period to distinguish the priority. The evaluation results can be used to purposefully assign scarce software and hardware resources to maximize defense gains.

The core of the effect evaluation is to design the corresponding evaluation algorithm according to the research question and the research object. An evaluation algorithm is a series of calculation steps that integrate multiple indicators into a total value. According to the input data, method attributes, and application category, we classified 15 classic evaluation methods in the DDoS evaluation field, as shown in Figure 1.

Univariate evaluation methods include index evaluation method (IEM) [5] and matrix analysis method (MAM) [6]. These methods are simple and have low computational
complexity. However, the one-sidedness of only one indicator determines that it cannot reflect all aspects of the evaluation object. Therefore, the multiattribute evaluation method is proposed to measure complex systems comprehensively. These multivariate evaluation methods include weighted sum/product method (WSM/WPM) [7], analytic hierarchy process (AHP) [8], and entropy method (EM) [9].

The above methods are all deterministic evaluation methods. They assume that all indicator data is available. When faced with the problem of uncertainty, the lack of ambiguous data conversion strategies makes these certainty methods unusable. Hence, uncertainty methods such as grey relational analysis (GRA) [10], rough set method (RSM) [11], and technique for order preference by similarity to an ideal solution (TOPSIS) [12] are proposed to quantify qualitative indicators. Besides, the evaluation method has also been improved to meet the needs of different applications. Deterministic evaluation methods such as WSM only require a set of data as input. It cannot make full use of the hidden relationships between different groups. In order to highlight the comparative significance of different results, evaluation methods that can process a numeric matrix in parallel have been proposed, including cluster analysis (CA) [13] and attack graph (AG) [14]. In uncertainty, manual data entry and configuration parameters are tricky issues. Aimed at this, fuzzy analysis (FA) [15] and system effectiveness analysis (SEA) [16] use fuzzy synthesis operators and trajectory domain to implement the fuzzy processing of system effectiveness.

The above evaluation methods provide calculation formulas that integrate DDoS evaluation indicators through mathematical theories such as statistics and topology. Nevertheless, there are still three significant problems. The first problem is that the result of the DDoS evaluation method is only determined by the input data and is not associated with the evaluation object. This one-sidedness leads to current methods, especially deterministic ones, susceptible to erroneous data and noise. Moreover, these interferences often appear in live network attacks, making this problem even more prominent. For example, the correlation coefficient of GRA and the ideal solution of TOPSIS are calculated based on extreme values, but whether each input is necessary has not been verified. The second problem is that the calculation of DDoS evaluation methods usually requires manual input of additional parameters, reducing the objectivity and credibility of the evaluation results. For example, the weight of WSM and the comparison matrix of AHP both need to be manually assigned. The third problem is that the calculation of the DDoS evaluation method defaults that all input indicators are independent of each other. These evaluation methods lack correlation processing, leading to redundant calculations in the evaluation results. For example, excessive memory usage will degrade software performance. The poor performance of the software makes the response delay of the provided service rise synchronously. Suppose that memory usage and response delay are directly integrated without correlation processing. In that case, the attack effect will be amplified, and the evaluation result will be distorted.

In the past ten years, artificial intelligence technology has made tremendous progress [17]. Significantly, the emergence of deep learning has dramatically accelerated its development, and various intelligent applications have emerged one after another [18]. This emerging technology has given us a new solution for evaluating the effectiveness of DDoS attacks. Deep learning is based on multilayer neural networks to train large sample sets. It can automatically learn the relationship between input and output. The neural network-based DDoS evaluation method has the nature of solving the above three problems. For question one, the
evaluation no longer solely relies on the input data. The neural network connects input and output through a backpropagation algorithm. For question two, the neural network trains weights and bias values through gradient descent. The parameters are learned through training, thereby avoiding subjective weighting. For question three, the nonlinear activation function of the neural network can handle nonmonotonic relations. Moreover, structures such as the convolutional neural networks (CNN) implement feature conversion, so that the output variables remain independent of each other.

This paper proposes a new method for evaluating the effectiveness of DDoS attacks based on neural networks. First, the two relationships between features and targets are clarified by analyzing a typical DDoS feature generation tool. Further, feature selection is realized based on the combination of the Triplet network and long short-term memory (LSTM). Through the convolutional network, deep feature information can be mined. The improved attention model is used to grade different local features. Finally, the result is synthesized through the fully connected layer.

The method proposed in this paper is important because we are the first to study DDoS attack effect evaluation based on deep learning theory. The main contributions of this research are as follows:

(i) The deep learning architecture was used for the first time in the field of DDoS attack effect evaluation;
(ii) An improved triplet network based on distance entropy was used to achieve efficient feature selection for DDoS evaluation;
(iii) The introduction of a multi-head attention mechanism solved the problem of assigning the same weight to different features.

The rest of this paper is organized as follows. Section 2 introduces background knowledge and related research. Section 3 extracts the DDoS characteristics and gives the neural network structure that can be used to evaluate the effect of DDoS attacks. Section 4 deals with the experimental dataset and verifies the effectiveness of our proposed neural network structure for DDoS evaluation. We sum up the research of this paper in Section 5. Section 6 gives the shortcomings of this paper and future research directions.

2. Related Work

In the DDoS effect evaluation, many evaluation methods are constantly being researched and put forward. Due to its low computational cost, the single indicator evaluation method is still widely used in DDoS evaluation. Paper [19] measured single indicators such as throughput and response time and analyzed the changes of indicators under different attack flow rates. Paper [20] compared the three entropy methods for evaluating the effect of DDoS attacks by analyzing the distribution of IP addresses. Paper [21] defined the amplification factor by calculating the proportion of delayed packets. It also implements corresponding reflection DDoS attack evaluation based on different application layer protocols. Paper [22] analyzed the impact on Tor through two typical DDoS attacks and measured the attack effect based on time consumption.

In the network confrontation environment, DDoS attack methods are usually diverse, and the background traffic is complicated. A single indicator evaluation method cannot meet the needs of comprehensive analysis in this case. Therefore, multivariant evaluation methods have gradually become the mainstream research direction of DDoS evaluation methods. Paper [23] proposed an automatic analysis of network attack effects based on attack graphs. This method assigns the probability of attack success to edge nodes to measure the degree of global damage. Paper [24] established a network attack indicator set and used fuzzy mathematics to give measurement criteria at different levels. Paper [25] established an evaluation indicator based on Delphi. It also used AHP to realize the security evaluation of the network system. Paper [26] established an indicator system by analyzing network attacks’ benefit and cost functions. It also uses dynamic Bayesian networks to evaluate the impact of attacks on network nodes. Paper [27] proposed a new indicator weighting method and realized a multilevel feedback security evaluation system through the core algorithm.

To overcome the low credibility of a single evaluation method, scholars usually combine multiple methods to obtain more scientific evaluation results. Paper [28] combined the construction of the attack graph and the generation process of the Bayesian network to achieve different levels of network attack quantification methods. Paper [29] proposed a trust framework and defined the basic elements. In addition, the matrix analysis and clustering method were combined to achieve reliable DDoS evaluation. Paper [30] used the fuzzy AHP method to explain the relationship between the evaluation target and attributes and obtained the overall evaluation value based on TOPSIS. Paper [31] proposed a two-step evaluation method for DDoS attacks. First, use network traffic entropy to depict the distribution of data packets; second, merge the fuzzy idea with entropy to improve evaluation sensitivity.

By combining several evaluation methods, their advantages and strengths are inherited while avoiding defects, so that the accuracy and credibility of evaluation can be improved to a certain extent. However, how to choose the appropriate evaluation methods and how to achieve effective integration still require the subjective judgment of cybersecurity experts and data scientists. In addition, this strategy also needs to design various multiattribute fusion methods according to the changes of application scenarios. Its uncertainty limits the scope of application and cannot promptly respond to new types of attacks. The emergence of deep learning technology has given us new solutions for DDoS attack evaluation. This technique learns complex correlations between input and output variables through a multilayer neural network or perceptron and updates weights and biases through a backpropagation algorithm. In-depth correlation information is automatically extracted from the input features during this training process without
human intervention. This automated procedure, which relies entirely on objective values, increases the reliability and confidence of the evaluation. In this paper, combined with the scene characteristics and elements of DDoS evaluation, we focus on the feature extraction and deep learning framework to design the DDoS evaluation method.

3. Methodology

To evaluate the effect of DDoS attacks, we extract features based on CICFlowMeter. In addition, we design a deep neural network to obtain the metric value of the DDoS attack effect. As shown in Figure 2, our proposed evaluation system consists of three main stages. First, the flow features in the collected original flow data are extracted as numerical vectors in the feature generation stage. On this basis, a feature subset suitable for DDoS attack evaluation is further selected. The training phase aims to establish an attack effect evaluation model. It takes the multidimensional feature vector at a certain moment as input. Then, output the evaluation value of the attack effect at this time. At this stage, a targeted neural network structure is selected to analyze the characteristics of DDoS traffic. Moreover, the number of the neural network’s layers and the number of neurons in each layer are constrained. The accuracy measurement method of DDoS attack effect evaluation is proposed in the evaluation stage. Based on this, through comparison with other evaluation methods, the effectiveness of our proposed neural network-based DDoS attack effect evaluation method is verified.

3.1. Feature Extraction. This section first analyzes a commonly used DDoS feature generation tool to find ideas for optimizing data before training. Then, LSTM is used to capture the relationship of input features over time. Finally, we use the Triplet network to compute the two correlations between features and the evaluation object concurrently.

3.1.1. Feature Generation. CICFlowMeter is a commonly used tool for generating network traffic characteristics [32]. The well-known network security public datasets CIC-IDS2017, CIC-IDS2018, and CIC-DDoS2019 are collected based on this feature extraction tool [33]. Due to its extensive application and good evaluation performance, CICFlowMeter has become one of the most effective collection tools for DDoS feature generation.

This tool converts the collected pcap files into 77 measurable features according to the data distribution of the 5-tuple [source IP, source port, destination IP, destination port, transport protocol]. These 77 indicators can be divided into the flow, packet, mark, and segment according to the different fine-grained objects. The results are shown in Table 1.

The construction of the DDoS evaluation indicator system needs to meet two principles. One is that indicators should be strongly related to the evaluation object. For example, the flooding attack that uses large traffic as attack mode is a typical DDoS attack. This attack approach does not take advantage of the vulnerabilities in the three-way handshake protocol. The SYN or ACK sign in the acknowledgment packet is consistent with normal traffic. Therefore, the mark features “SYN Count” and “ACK Count” in Table 1 cannot reflect the attack effect of the flooding attack. If all indicators are used indiscriminately for effect evaluation, the results will contain more external interference. The second is to minimize the redundancy between indicators. From Table 1, we can see that similar features exist in a feature subset. For example, “Fwd IAT Std” and “Fwd IAT Mean” describe the flow interval, and “Packet Length Variance” and “Packet Length Mean” describe the packet length. There is a standard deviation and variance function relationship between them. Assume that this association is not considered. In this case, it will cause the implicit double calculation in the total value, affecting the accuracy of the final evaluation result.

The dataset labels can guide supervised training and measure the neural network’s performance on the test set. DDoS attack effect evaluation belongs to supervised learning, so a good label is crucial. CICFlowMeter only emphasizes basic flow characteristics and ignores the generation of corresponding labels. Paper [34] defines a new label generation method. It is proposed that, under a sufficiently large sample, the attack effect can be estimated by the attack duration and the number of data packets. DDoS attacks have enormous data packets, which satisfies the premise. More importantly, in addition to the forward attack flow directly sent by the attacker, the DDoS attack also includes the backward attack flow generated by the reflective amplifier. Therefore, we utilize the three features “Flow duration,” “Total Fwd Packet,” and “Total Bwd Packet” to calculate the attack effect label. Its calculation formula is as follows:

\[
\text{label} = \text{duration} \cdot \text{packet}_{\text{fwd}} \cdot \text{packet}_{\text{bwd}}.
\]

We do not use equation (1) but a comprehensive algorithm to measure the effect of DDoS attacks. The reason is that, in a statistical sense, equation (1) can only be approximated when there are enough sample points. It would lose efficacy in a small interval. In the early attack stage, realizing quick attack effect measurement based on small data can gain more prime time for subsequent security strategy deployment. Therefore, we aim to find a way to quickly measure small samples’ attack effect through a comprehensive analysis of large samples (such as neural networks).

3.1.2. Feature Selection. Feature selection is a necessary prestep for efficient neural network training [35]. According to Section 3.1.1, CICFlowMeter can generate a complete set of DDoS features. Dataset optimization can be achieved by connecting feature selection afterward. It achieves the filtering of irrelevant features by adjusting the objective function. It also reduces the probability of redundancy by controlling the number of features and speeds up the convergence of the neural network. The current DDoS feature selection depends on the empirical judgment of
network security administrators. This method is not objective, easy to prioritize nonserious cyber attacks due to inevitable subjective errors. Consequently, a vicious attack will lose the best time to handle it.

Let $st$ denote the relationship function between features and objects, and let $re$ denote redundancy between features. Then, the best indicator system can be expressed as follows:

$$\text{Best_subset} = \arg\text{Subset} \left( \max \left( st \left( \text{subset} \right) \right), \min \left( re \left( \text{subset} \right) \right) \right).$$

Triplet network (TN) is an improved version of the Siamese network (SN) [36, 37]. SN implements the similarity measurement of two features in the same set of inputs. It cannot calculate the relationship among three or more features. TN expands SN’s two parallel sample inputs to three, thereby modeling the relationship between multiple features. The three inputs can be expressed as $\{\text{anchor sample } x, \text{positive sample } x^+, \text{negative sample } x^-\}$. Unlike conventional neural networks that rely on labels, this model is an unsupervised learning network. Its training depends on the comparison between features. A set of feature inputs is essentially a time vector, and its time variation pattern can be obtained through LSTM [38]. Inspired by the TN model, we proposed an improved model, namely, LSTM-based Triplet network (LTN), to achieve multiple correlation analyzes. We use positive sample $x^+$ to simulate object $label$, anchor sample $x$ to simulate candidate indicator $i$, and negative sample $x^-$ to simulate remaining indicator $\bar{i}$. The LTN model is shown in Figure 3.

In Figure 3, dataset $S$ is a real matrix with $m$ features as rows and $n$ moments as columns. Among inputs of LTN, feature set $i = \{i_1, i_2, \ldots, i_m\}$, time set $t = \{t_1, t_2, \ldots, t_n\}$, and $d_{pq}$ denotes the value of feature $t_p$ collected at time $t_q$, label = $\{l_1, l_2, \ldots, l_n\}$ is the ideal value for evaluating the effect of DDoS attacks at different times. Specifically, feature set $i$ can be collected by a feature generator tool, such as CICFlowMeter, or designed subjectively by researchers. Time set $t$ is the collection time interval sequence. Object label can be obtained by mapping existing labels in the public dataset or approximated by equation (1). Among the three neural network inputs in Figure 3, evaluation object label is the reference effect values, fixed as the label sequence unchanged. Candidate indicator $i$ is randomly assigned from the feature set. Remaining indicator $\bar{i}$ is selected from other features except $i$. Notably, the disorder indicator pair $\langle i, \bar{i} \rangle$ cannot be calculated repeatedly. Thus, there are a total of $C_n^2$ indicator pairs. After $C_m^2$ iterations, the feature selection is completed.

The three features simultaneously input the same neural network structure and share weights and bias values in one iterative calculation. Each feature is trained separately through a neural network and does not interfere with each other. The input indicator is essentially a long-time series vector. Therefore, to find the complex relationship between different moments, we use a three-layer LSTM as the neural network structure. In order to coordinate multiple iterations, a triplet loss node is added at the end of LTN. This node updates the gradient descent process through backpropagation. Triplet loss aims to achieve the smallest redundancy between features and the strongest association with the target concurrently in multiple iterations. For triplet (label, $i$, $\bar{i}$), the calculation formula of triplet loss is expressed as

$$\text{Triplet Loss} = \max \left( d \left( i, label \right) - d \left( i, \bar{i} \right) + \ell, 0 \right),$$

where $d \left( i, label \right)$ and $d \left( i, \bar{i} \right)$ are the Euclidean distances from candidate indicator $i$ to object $label$ and another indicator $\bar{i}$, and $\ell$ is the minimum distance between these two distances. Empirically speaking, $\ell$ is usually set to 0.6 [39], and $d \left( \ast, \ast \right)$ uses the softmax function to convert its value between 0 and 1. However, this distance is based on the solution of the spatial distance. It represents the linear distance between two sample points in a multidimensional space. It cannot reflect the coordinated change of the two. Aiming at this problem, we introduce the distance entropy of order $a$ to mirror the difference in the overall system distribution. This distance can measure monotonic correlation. The calculation formula of $D \left( i, label \right)$ is as follows:
\[
D(i, \text{label}) = \frac{1}{\alpha - 1} \log_2 \left( \sum_n \left( \frac{d_{mn}}{\sum_n d_{mn}} \right)^a \left( \frac{\text{label}_n}{\sum_n \text{label}_n} \right)^{1-a} \right).
\]

Similarly, the calculation formula of \(D(i, \tilde{i})\) is as follows:

\[
D(i, \tilde{i}) = \frac{1}{\alpha - 1} \log_2 \left( \sum_n \left( \frac{d_{mn}}{\sum_n d_{mn}} \right)^a \left( \frac{\tilde{i}_n}{\sum_n \tilde{i}_n} \right)^{1-a} \right).
\]

3.2. Structure of Deep Learning Model. We proposed a novel neural network-based DDoS evaluation (NNDE) to measure the DDoS attack effect. It includes four main parts, namely, embedding layer, convolutional layer, attention model, and fully connected layer. The overall structure of NNDE is plotted in Figure 4. Besides, we explain each part’s selection basis and implementation details in the following subsections.

3.2.1. Embedding. The initial feature set is obtained by simplifying the original dataset according to the feature selection model in Section 3.1.2. Compared with the original dataset, the number of features in the initial dataset is reduced. Each feature is a numeric vector collected under a multidimensional time series. What is more, the long-term confrontation of DDoS attacks makes this time series a high dimension. For example, in the CIC-DDoS2019 test set, the traffic items of TFTP reached 20,107,827. High-dimensional data makes DDoS evaluation a curse of dimensionality [40]. It will be challenging to converge if it is directly input to the feedforward neural network without further data processing. Therefore, to facilitate subsequent neural network training, the reduced feature set will first pass through an embedding layer to reduce dimensionality. Each state vector of the embedding layer is obtained by calculating the relationship between two features in the feature set. This relational calculation can be mapped from a high-dimensional sparse matrix to a low-dimensional dense matrix. All vectors in the embedding layer constitute the embedding matrix. When the matrix size is \(64 \times 1000\), it can meet the dimensionality reduction requirements of most network security datasets.

3.2.2. Convolutional Layer. The convolution part comprises a two-dimensional convolutional neural network (2D-CNN) and a max-pooling layer. The output of the embedding layer is still a numerical matrix, so we use 2D-CNN rather than a one-dimensional convolutional neural network (1D-CNN) for training. The 2D-CNN layer has two functions. First, the
3.2.3. Attention Model. The convolutional layer focuses on the weight of a single feature. However, it ignores the comparison of local features, so that the learned different local features have the same importance. Consequently, the impact of multiple local features on the network attack effect cannot be well measured. We introduced a multihead attention model to add weight to different local features and eliminate the influence of repeated features [41]. The structure of the multihead attention model is shown in Figure 5. The idea of attention comes from the automatic coder-decoder [42], which calculates the input of the next moment through the weighted average of the implicit correlation between current outputs. We used the multihead attention mechanism to capture the interaction between different local features. We also modified the attention model to better deal with DDoS characteristics. We added ResNet to reconstruct the intermediate data and 1D-CNN for information clustering and parameter reduction [43]. The reason for choosing 1D-CNN here is that the feature matrix is segmented and separately calculated after ResNet and Add&Norm.

In Figure 5, the attention model includes three main parts: multihead attention, residual and normalization, and 1D-CNN. Among them, multihead attention integrates the deeply hidden information learned by the model in multiple expression subspaces [44]. Let $i^u_i$ be the $u$-th indicator in the indicator set $i$. Take the parameter transfer process of $i^u_i$ in each subspace to illustrate information flowing in the attention model. Suppose that the total number of indicator heads is $R$, then arbitrary indicator head $r \in \{1, 2, \ldots, R\}$. Let $\delta'^*(i^u_i, i'^v_r)$ denote the correlation between indicator $u$ and indicator $v$, and let $\eta_{u,v}^r$ denote the correlation coefficient between indicator $u$ and indicator $v$. The relationship between the two is as follows:

$$
\eta_{u,v}^r = \frac{\exp(\delta'(i^u_i, i'^v_r))}{\sum_{m=1}^m \exp(\delta'(i^u_i, i'^v_m))}
$$

(7)

In equation (7), the calculation method of $\delta'(i^u_i, i'^v_r)$ includes dot product and perceptron. The dot product has a less computational cost, so we use it to converge quickly. Its formula is as follows:

$$
\delta'(i^u_i, i'^v_r) = \langle Q^r i^u_i, K'^r i'^v_r \rangle, Q'^r, K'^r \in \mathbb{R}^{\delta \times s'}
$$

(8)

where $Q'$ and $K'$ represent the key-value pair $\langle \text{Query}, \text{Key} \rangle$ of the input indicator $i'^v_r$. This key-value pair is the conversion parameter that maps indicator $i'^v_r$ from the initial sample space $\mathbb{R}^s$ to the new sample space $\mathbb{R}^{\delta \times s'}$. The essence of equation (8) is to calculate the correlation between the queue and key. Its result is used as the attention weight. The calculation formula for the indicator head $r$ of the indicator $i'^v_r$ is as follows:

$$
\overline{i'^v_r} = \sum_{j=1}^m \eta_{u,j}^r (V'^r i^u_i), V'^r \in \mathbb{R}^{\delta \times s'}
$$

(9)

The mapping of the indicator $i'^v_r$ in the new sample space can be formed by concatenating all the conversion formulas $\overline{i'^v_r}$ in the indicator header set. The result is as follows:

$$
\overline{i'^v} = \text{Concat}(\overline{i'^v_1}, \ldots, \overline{i'^v_R})
$$

(10)

Up to now, we have regenerated the weights of different local features. In the above process of generating the new space solution, the calculation of different indicator heads is parallelized. However, it is prone to decay in training due to more network layers and is difficult to converge. To solve this, we use ResNet to optimize the network structure. It makes the training process more sensitive to parameter
changes, improves the efficiency of forward propagation, and solves the disappearance of backward propagation gradients. After N-layer ResNet, the change of indicator $i^u$ can be expressed as

$$i^u = i^n + \sum_{j=1}^{N-1} \text{ReLu}\left(\frac{i^j}{w_j}\right),$$

where $w_j \in \mathbb{R}^{s \times s}$ denotes a transform matrix composed of the $j$th layer network’s parameters, and ReLu is a nonlinear activation function. After ResNet, they are connected to the normalization layer to make the gradient descent training process smoother and improve the generalization ability. Under the output $\{i^j, 1 \leq j \leq m\}$ of different features, the overall output $\hat{T}$ of the attention model can be expressed as

$$\hat{T} = \text{Concat}(\hat{i}^1, \ldots, \hat{i}^m).$$

3.2.4. Fully Connected Layer. The fully connected layer uses neurons to perform nonlinear transformations on the output of the attention model to obtain the predicted value of the DDoS attack effect. Commonly used nonlinear activation functions include sigmoid, tanh, and ReLu [45]. Among them, the asymmetric expression of sigmoid makes its output not centered on the origin. This deviation causes the weight to tilt to the positive axis during the deep gradient descent, affecting the comparative meaning of the difference in attack effect evaluation at different points. Tanh takes exponential calculation as the core. Its backpropagation slope is low and tends to the problem of gradient disappearance, which leads to a slow learning process. ReLu takes
the origin as the center of rotation and alleviates the interference of non-zero symmetry on the result to some extent. This function is based on the linear operation, so the calculation rate is fast. Further, the oversaturation problem can be effectively solved by adjusting the learning rate. Therefore, we use ReLu as the activation function.

The number of fully connected layers usually appears as a hyperparameter. Besides adjusting the parameters during the experiment, we also try to explain the meaning of the number of layers theoretically. For this reason, we compared the application scenarios, advantages, and disadvantages of fully connected networks with different layers, as shown in Table 2.

DDoS attack effect evaluation is a regression problem. From Table 2, we can see that the regression solution requires that the number of fully connected layers is not less than two. In addition, its number is not less than three to learn deeper complex features. However, converging in more than four fully connected layers is challenging. Combining the above restrictions, we set the number of fully connected layers to three, each with 16 units. Unlike the classification problem, the activation function of the last layer is not softmax but sigmoid, which generates an actual value between zero and one. Its size indicates the effectiveness of the attack. A dropout layer is connected behind the convolutional and fully connected layers to prevent overfitting. Let \( w \) and \( b \) denote the weight and bias of the fully connected layer, respectively. The final attack effect value at time \( t \) can be expressed as

\[
T_t = \text{Sigmoid}(WT^t + b), W \in \mathbb{R}^{1 \times 13}. \quad (13)
\]

4. Evaluation

In the field of network security, commonly used DDoS datasets include KDD99 [46, 47], NSL-KDD2009 [48], CIC-IDS2017 [49], CIC-IDS2018 [50], and CIC-DDoS2019 [51]. The KDD99 and NSL-KDD2009 datasets are usually used as benchmark datasets for DDoS attack effect evaluation. The CIC-IDS2017, CIC-IDS2018, and CIC-DDoS2019 datasets have overcome the shortcomings of the benchmark datasets, such as unextracted reverse features and uncaught reflection DDoS attacks. These three datasets have become the most convincing representatives of network security datasets. To improve the experiment’s credibility, we chose these five well-known DDoS public datasets as the experimental datasets.

The experimental part includes the validity verification of feature selection and the validity verification of the proposed neural network. The design logic is shown in Figure 6. It can be seen from Figure 6 that the necessity and effectiveness of feature selection are first verified. Secondly, the effectiveness of DDoS attack evaluation based on the proposed neural network is verified under the premise of feature selection. The comparative experiment includes four basic units: dataset, neural network, evaluation algorithm, and feature selection method. The connection between these four units and an experiment indicates that they will be considered in the current experiment. For example, the link marked \( \circ \) connects the neural network and evaluation dataset to the first experiment, indicating that this experiment needs to consider the influence of different datasets and network structures.

Specifically, in the comparison experiment of feature selection methods, we selected three standard statistical feature selection methods and four classical neural networks as the comparison objects. Three statistical methods include chi-square (CHI) [52], relevant feature selection (Relief) [53], and information gain (IG) [54]. The four classic neural network structures include deep neural networks (DNN), CNN, recurrent neural networks (RNN), and CNN + RNN. Besides the neural network structure, four common evaluation methods are selected as the comparison objects in the comparison experiment of the attack effect evaluation, namely, certainty evaluation WSM, uncertainty evaluation GRA and TOPSIS, and combined evaluation GRA + TOPSIS.

It is worth mentioning that, for clarification, we did not compare the method proposed in this paper with popular DDoS detection methods. As we all know, both DDoS detection and DDoS assessment can be analyzed and studied through deep learning methods. However, in essence, DDoS detection is a classification problem, and DDoS evaluation is a prediction problem. There are great differences between the two in terms of research purpose, realization method, and parameter design. For example, DDoS detection distinguishes attack traffic from normal traffic through traffic characteristics, and DDoS evaluation quantifies the strength of an attack by integrating evaluation indicators. The former finally gets 0 or 1 to mark whether it has suffered a DDoS attack, while the latter’s calculation result is a scalar that can take an arbitrary floating-point number, which represents the strength of the attack the defender is experiencing. This scalar contains additional information that cannot be expressed simply by solving the binary classification problem. Therefore, to highlight the difference between the two, we did not compare the state-of-the-art methods for DDoS detection with the NNDE evaluation method proposed in this paper.

4.1. Data Processing. First, we extracted DDoS-related attack traffic from five datasets. The number is shown in Table 3. Notably, the NSL-KDD2009 dataset has added the defense difficulty label. The larger this number, the easier it is to detect the attack, and the lower the attacking threat, and conversely, the higher the threat of attack. This label can be used to indicate the attack effect. Differently, the other four datasets are unlabeled. So, we manually added the label according to Eq. 1.

Noteworthy, neural networks and evaluation algorithms require numeric vectors as input and cannot handle string data. These string data include two types, namely, numbers with no comparative meaning and undefined special characters. For the first type, such as traffic ID and source IP, we deleted them from the dataset.
For the second type, we replaced "Infinity" with statistical values (such as extremum) and removed traffic data "NaN."

Since the dimensions and range of indicators are different, it is senseless to synthesize them directly. So, indicators must be standardized before the evaluation begins. There are three main normalization methods: StandardScaler, MinMaxScaler, and RobustScaler [55]. Among them, the calculation of mean and standard deviation in StandardScaler and the extreme value in MinMaxScaler are easily affected by outliers, affecting the comparison of intermediate values. In contrast, RobustScaler uses quantiles as a reference, so a single value does not affect this method. To sum up, we chose RobustScaler as the normalization method to reduce errors. Its formula is as follows:

\[
d_{pq}^\prime = \frac{d_{pq} - \text{median}}{\text{quantile}_{25} - \text{quantile}_{25}}
\]

where median denotes the mean value, quantile_{25} denotes the upper quartile, and quantile_{25} denotes the lower quartile.

4.2. Measurement. The validity verification of our proposed evaluation method includes the comparison with mainstream neural networks and commonly used evaluation algorithms. In the comparative experiment of neural networks, the minimum mean square error (MSE) [56] does not signify that the evaluation value is the most accurate. It is also essential to assess whether the results are evenly distributed. Therefore, in addition to MSE, we also use the explained variance score (EVS) [57] to investigate the discrete distribution of results. Their formulas are as follows:

\[
\text{MSE} = \frac{1}{n} \sum_{j=1}^{n} (T_j - \text{label}_j)^2,
\]

\[
\text{EVS} = 1 - \frac{\text{Var}(T_j - \text{label}_j)}{\text{Var}(T_j)}
\]

where \(n\) denotes the sampling time, \(T_j\) and \(\text{label}_j\) denote the estimated value and label value at the \(j\)th time, respectively, and \(\text{Var}(\ast)\) denotes the variance calculation. Because the calculation principles are not the same, different evaluation algorithms will usually get distinct values for the same data input. In the comparison experiment with other evaluation algorithms, common measures MSE and MAE were not selected. The reason is that both calculate the spatial distance between single points and lack comparison. Further, we put forward a new measure Sort\_Acc to compare effects at different moments. It is defined as follows:

\[
\text{Sort\_Acc} = \frac{\text{num}(\hat{i}^n \cap \text{label})}{\binom{n}{2}},
\]

where \(\text{num}(\hat{i}^n \cap \text{label})\) denotes the intersection of ordered pairs in \(\hat{i}^n\) and \(\text{label}\), and \(\binom{n}{2}\) denotes the number of all possible ranking pairs. In equation (16), there are three relationships: greater than, less than, and equal to. The non-equal relationship between two labels is judged whether the comparison relationship of the corresponding predictive values is consistent with it. If equal, Sort\_Acc will increase by one. The equality relationship between two labels is judged whether the difference between the corresponding outputs exceeds 10% of the smaller value. If less, the two values are equal, and Sort\_Acc will increase by one. Otherwise, it is not.

### Table 2: Comparison of different fully connected layers.

| Num | Applicable scene                          | Disadvantages                                      | Advantages                             |
|-----|------------------------------------------|----------------------------------------------------|----------------------------------------|
| 0   | Linear, simple classification            | Cannot cope with complex relationships             | Simple, fast convergence               |
| 1   | Nonlinear, simple classification         | Low accuracy                                       | Continuous function can be expressed   |
| 2   | Nonlinear, classification and regression | Limited learning characteristics                    | Smooth mapping can be expressed        |
| 3   | Nonlinear, classification and regression | Slow convergence                                   | Complex features can be learned        |
| ≥ 4 | Nonlinear, classification and regression | Very slow convergence                               | Complex features can be learned        |

### Table 3: The number of DDoS-related traffic in five datasets.

| Dataset      | KDD99 | NSL-KDD2009 | CIC-IDS2017 | CIC-IDS2018 | CIC-DDoS2019 |
|--------------|-------|-------------|-------------|-------------|--------------|
| DDoS traffic | 4856151 | 45927       | 128027      | 687742      | 50063112     |

### Figure 6: Two of the validation subexperiment design logic.
In this part, according to the different comparison objects, we choose different measurement strategies to illustrate the quality of the attack effect evaluation. When the proposed NNDE method is compared with other neural network methods, the commonly used metrics MSE and EVS can cope with this situation. However, the same input usually results in greatly different output values for different statistical evaluation methods due to the variable calculation principles. In this case, MSE or EVS based on numerical regression cannot be used for comparison. Although the obtained values are different, the order of the different evaluation values can still reflect the quality of the evaluation results. Based on this, when comparing with statistical methods, we proposed a new metric named Sort Acc to uniformly describe the impact of evaluation methods on ranking evaluation results.

4.3. Result and Analysis. We built a Keras-based neural network test platform to verify the effectiveness of our proposed method. TensorFlow is used as the backend training engine. It supports modular network structure programming and can execute on a CPU or GPU. What is more, the parallel computing capabilities of GPUs have greatly improved the training speed of deep neural networks.

The configuration parameters of the workstation running the experiment are 64G RAM, 1TB solid-state disk, i5-9300HF processor, and GeForce GTX 1660Ti graphic card. Since the number of normal samples in the dataset is much larger than the number of attack samples, we use k-fold cross-validation to divide the original dataset. Concretely, in the experiment, the value of k is set to 10. In each training, one share is selected as the test set, and the remaining nine shares are used as the training set.

4.3.1. Validation Experiment of Feature Selection. First, we examined the necessity of feature selection for DDoS attack effect evaluation. No feature selection was used as the control group, and feature selection was used as the experimental group. We compared the difference between the two in terms of accuracy and running time. Specifically, CHI, Relief, and IG are used as the three feature selection methods of the experimental group, and CNN and RNN are used as the two evaluation neural networks. Train ten times and take the average to reduce errors. The verification results on the five datasets are shown in Figure 7. It can be seen from Figure 7(a) that the DDoS attack effect evaluation with feature selection has achieved a higher accuracy rate than that without feature selection. Especially on the NSL-KDD2009 dataset, the accuracy rate after feature selection increases from 44.1% to 75.3%, and the improvement rate reaches 31.2%. Figure 7(b) shows that feature selection can remarkably reduce the training time. When the amount of data is larger, the decrease rate in time consumption is greater. On the CIC-DDoS2019 dataset, the highest average consumption time reduction rate is 28.5%. In short, feature selection can improve training efficiency and training accuracy at the same time and is an essential part of efficient and high-performance effect evaluation.

Secondly, the effectiveness of our proposed LTN model was verified by comparing it with other neural network structures and statistical feature selection methods. We reproduced the traditional feature selection methods and ran them on the CPU. Deep learning methods were run on GPU and CPU separately to compare time consumption. The comparison includes the accuracy of ranking pairs and the training time of evaluation. The average of multiple pieces of training is taken. K99, N09, C17, C18, and C19 represent five datasets. The results are shown in Table 4. It can be seen from Table 4 that the LTN method has reached the highest sorting accuracy on the five datasets, respectively, 93.1%, 92.3%, 92.9%, 93.7%, and 94.6%. The statistical-based feature selection method has the lowest accuracy rate, and its highest accuracy rate does not exceed 74%. In addition, the combination of neural networks has achieved better performance than a single network. However, there is still an average gap of 5.52% from the LTN method. From Table 4, we can see that, in addition to DNN, deep learning methods converge faster than traditional statistical methods. In particular, the LTN method achieves the fastest convergence rate on five datasets. The deep learning method running on GPU during the training process is about 16 times more efficient than the CPU. The DNN structure has many parameters, and statistical methods cannot use GPU acceleration, making it difficult for both to process large-scale data.

All in all, the comparison with the other seven feature selection methods verifies the effectiveness of the LTN method in selecting DDoS evaluation features. This method can better reduce the training time and improve evaluation accuracy. In subsequent experiments, the feature selection method is set to LTN.

4.3.2. Validation Experiment of Neural Network Evaluation. First, we verified the effectiveness of NNDE for DDoS effect evaluation under the comparison of four neural networks. Specifically, the single DNN or CNN structure maintains the same parameter settings as the NNDE network to control variables. For RNN, we use a three-dimensional LSTM structure of 32 neurons to represent it. MSE and EVS are used to measure neural network regression error and distribution, respectively. The comparison result is shown in Figure 8. It can be seen from Figure 8 that the NNDE method achieves the smallest regression error and the highest distribution stability on the five datasets. From Figure 8(e), our proposed NNDE method performs best on the CIC-DDoS2019 dataset. Its average error is 0.27, and the variance score is 0.82. In particular, as shown from Figure 8(b), on the NSL-KDD2009 dataset with small samples, NNDE still controls the average error within 0.45, and the variance number exceeds 0.7, showing a robust fitting ability.

Secondly, we also verified the effectiveness of NNDE under the comparison of four classic evaluation algorithms. The Python-based Pandas library implements the parallel matrix calculation of the evaluation algorithm. Specifically, WSM averages multiple weight calculations
Table 4: Comparison of LTN and other feature selection methods.

| FS method | Sort_Acc | Running time |
|-----------|----------|--------------|
|           | K99 | N09 | C17 | C18 | C19 | K99 | N09 | C17 | C18 | C19 |
| DNN       | 0.768  | 0.797 | 0.826 | 0.802 | 0.811 | 56.41 | 25.11 | 31.61 | 40.32 | 2482.29 |
| CNN       | 0.813  | 0.785 | 0.812 | 0.834 | 0.835 | 19.62 | 14.43 | 15.35 | 16.88 | 789.53 |
| RNN       | 0.790  | 0.763 | 0.829 | 0.836 | 0.842 | 21.14 | 16.06 | 16.99 | 20.71 | 861.81 |
| CNN + RNN | 0.877  | 0.854 | 0.871 | 0.874 | 0.914 | 17.06 | 14.26 | 14.86 | 19.98 | 738.91 |
| CHI       | 0.671  | 0.652 | 0.664 | 0.683 | 0.735 | 38.17 | 23.57 | 28.12 | 34.63 | 1286.26 |
| Relief    | 0.646  | 0.637 | 0.64  | 0.631 | 0.692 | 77.69 | 45.22 | 56.73 | 61.93 | 3187.92 |
| IG        | 0.662  | 0.641 | 0.673 | 0.682 | 0.726 | 50.93 | 25.06 | 30.28 | 35.93 | 1917.64 |
| LTN       | 0.931  | 0.923 | 0.929 | 0.937 | 0.946 | 11.24 | 8.23  | 9.3  | 10.86 | 438.12 |

Figure 7: The necessity of DDoS feature selection in effect evaluation.

Figure 8: Continued.
Figure 8: Performance comparison between NNDE and standard neural networks. (a) KDD99, (b) NSL-KDD2009, (c) CIC-IDS2017, (d) CIC-IDS2018, and (e) CIC-DDoS2019.

Figure 9: Continued.
to reduce the subjective influence. The grey correlation coefficient of GRA is set to 0.5, and TOPSIS is estimated based on the distance from the label to the best and worst solutions. Unlike other neural network structures, in this experiment, Sort_Acc is used to measure the difference between the statistical evaluation method and the NNDE method. The evaluation results on the five datasets are shown in Figure 9. It can be seen from Figure 9 that the NNDE method achieves the highest accuracy rates on all datasets, which are 87.2%, 91.3%, 88%, 85.6%, and 94.5%, respectively. Compared with WSM and combined evaluation methods, the accuracy rates are increased by 21.96% and 7.12%, respectively. Overall, NNDE has achieved an average accuracy improvement of 19.73% compared with other evaluation methods.

Finally, we compared the running time of NNDE with other neural network structures. Each neural network runs ten times, and the average time consumption is shown in Table 5. It can be seen that the more complex the network structure, the more the training parameters, and the longer the running time. Compared with the combined network structure, the time consumption of NNDE has increased by 18.02%. Although the time complexity increases during training, this part of the overhead has no negative impact on our evaluation model application. The reason is that the structure and parameters of the deep learning model are preserved once pretraining is done. Evaluation can be performed using existing models in subsequent evaluations without repeating complex training. Therefore, this added time overhead does not interfere with real-time evaluation.

In short, the comparison with four commonly used neural network structures and four classic evaluation algorithms verifies the effectiveness of the proposed NNTE method for DDoS attack effect evaluation. The highest accuracy rate reaches 94.5%. Time expenditure is controlled within a reasonable range to facilitate subsequent defense strategies’ rapid deployment.

Table 5: Comparison of running time between NNDE and four neural network structures.

| Network structure | DNN | CNN | RNN | CNN + RNN | NNDE |
|-------------------|-----|-----|-----|-----------|------|
| Running time      | 17.83 | 21.93 | 24.92 | 36.74 | 43.36 |

Figure 9: Performance comparison between NNDE and statistical evaluation methods. (a) KDD99, (b) NSL-KDD2009, (c) CIC-IDS2017, (d) CIC-IDS2018, and (e) CIC-DDoS2019.
5. Conclusion

This paper proposes a new method for evaluating the effectiveness of DDoS attacks based on neural networks. We aim to solve the low accuracy of traditional effect evaluation due to the lack of feature selection and targeted evaluation methods. Based on the feature generation tool CICFlowMeter, the redundancy and correlation issues in the DDoS features are analyzed. The relationship between multiple features is captured by combining the TN network and LSTM. Moreover, distance entropy is used to realize the difference measurement of the distribution. In constructing the evaluation neural network, the local features of the vector are captured based on the convolutional layer. The improved attention model achieves hierarchical training of different local features. Finally, feature aggregation is achieved through a fully connected layer. The experimental results show that our proposed neural network-based DDoS attack effect evaluation method could effectively improve evaluation accuracy. More importantly, it provides basic support for the designation of defense plans by security experts.

6. Future Research

We verified the effectiveness of our proposed method from both theoretical and experimental aspects. However, there are still the following problems to be solved.

**Question 1.** In this article, we generated labels according to equation (1). This method aims at large traffic attacks. Due to similarity with normal traffic, slow DDoS attacks cannot be labeled. So, we need to find a new way to label slow DDoS attacks in the future.

**Question 2.** In the experiment, we extracted DDoS-related traffic to verify the proposed ideas. Whether this method can apply to other traffic types still needs further verification.

Data Availability

The KDD99, NSL-KDD2009, CIC-IDS2017, CIC-IDS2018, and CIC-DDoS2019 datasets used to support the finding of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Thanks are due to the artificial intelligence research team of the State Key Laboratory of Mathematical Engineering and Advanced Computing for their help and encouragement. Especially, Dr. Qiu proposed meaningful revisions for the experiments in this article.

References

[1] G. S. Kushwah and V. Ranga, “Voting extreme learning machine based distributed denial of service attack detection in cloud computing,” *J. Inf. Secur. Appl.*, vol. 53, 2020.

[2] T. Ehrenkranz and J. Li, “On the state of IP spoofing defense,” *ACM Transactions on Internet Technology*, vol. 9, no. 2, 2009.

[3] C. Fachkha, E. Bou-Harb, and M. Debbabi, “Inferring distributed reflection denial of service attacks from darknet,” *Computer Communications*, vol. 62, 2015.

[4] S. Behal, K. Kumar, and M. Sachdeva, “D-FAC: a novel ϕ-Divergence based distributed DDoS defense system,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, 2018.

[5] H. yuan Fang, S. wei Gan, and C. ying Xue, “Evaluation of regional water resources carrying capacity based on binary index method and reduction index method,” *Water Sci. Eng.*, vol. 12, no. 4, 2019.

[6] W. Wei, L. Zhang, Y. Jiao, C. Tian, C. Wang, and Y. Zhang, “Intracluster structured low-rank matrix analysis method for hyperspectral denoising,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, 2019.

[7] J. R. S. C. Mateos, “Weighted sum method and weighted product method,” *Green Energy and Technology*, vol. 83, 2012.

[8] A. Darko, A. P. C. Chan, E. E. Ameyaw, E. K. Owusu, E. Pärn, and D. J. Edwards, “Review of application of analytic hierarchy process (AHP) in construction,” *Int. J. Constr. Manag.*, vol. 19, no. 5, 2019.

[9] L. Chen, S. Gao, B. Liu, Z. Lu, and Z. Jiang, “FEW-NNN: a fuzzy entropy weighted natural nearest neighbor method for flow-based network traffic attack detection,” *China Commun.*, vol. 17, no. 5, 2020.

[10] M. Yazdani, C. Kahraman, P. Zarate, and S. C. Onar, “A fuzzy multi attribute decision framework with integration of QFD and grey relational analysis,” *Expert Systems with Applications*, vol. 115, 2019.

[11] Y. Qian, X. Liang, G. Lin, Q. Guo, and J. Liang, “Local multigranulation decision-theoretic rough sets,” *International Journal of Approximate Reasoning*, vol. 82, 2017.

[12] P. Chen, “Effects of the entropy weight on TOPSIS,” *Expert Systems with Applications*, vol. 168, 2021.

[13] M Benassi, “Using two-step cluster Analysis and latent class cluster Analysis to classify the cognitive heterogeneity of cross-diagnostic psychiatric inpatients,” *Frontiers in Psychology*, vol. 11, 2020.

[14] H. S. Lallie, K. Debattista, and J. Bal, “A review of attack graph and attack tree visual syntax in cyber security,” *Computer Science Review*, vol. 35, 2020.

[15] J. F. Chen, H. N. Hsieh, and Q. H. Do, “Evaluating teaching performance based on fuzzy AHP and comprehensive evaluation approach,” *Applied Soft Computing J.*, vol. 28, 2015.

[16] N. Polatidis, M. Pavlidis, and H. Mouratidis, “Cyber-attack path discovery in a dynamic supply chain maritime risk management system,” *Computer Standards & Interfaces*, vol. 56, 2018.

[17] D. Gunning and D. W. Aha, “DARPA’s explainable artificial intelligence program,” *AI Mag.*, vol. 40, no. 2, 2019.

[18] L. Liu, W. Ouyang, X. Wang et al., “Deep learning for generic object detection: a survey,” *International Journal of Computer Vision*, vol. 128, no. 2, 2020.

[19] M. Sachdeva, K. Kumar, G. Singh, and K. Singh, “Performance analysis of web service under DDoS attacks,” in *Proceedings of the IEEE Int. Adv. Comput. Conf. IACC 2009*, pp. 1002–1007, Patiala, India, February 2015.
[20] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection," *Pattern Recognition Letters*, vol. 51, 2015.
[21] B. Siewklik, R. MacFarlane, and W. J. Buchanan, "Evaluation of TFTP DDoS amplification attack," *Computers & Security*, vol. 57, pp. 67–92, 2016.
[22] B. Conrad and F. Shirazi, "Analyzing the effectiveness of DoS attacks on tor: [fast abstract]," in *Proceedings of the ACM International Conference Proceeding Series*, September 2014.
[23] O. Shoeyber, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, " Automated generation and analysis of attack graphs," in *Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, May 2002.
[24] X. Fan, C. Li, and Y. Wang, "Strict intuitionistic fuzzy entropy and application in network vulnerability evaluation," *Soft Comput*, vol. 23, no. 18, 2019.
[25] M. K. Chen and S. C. Wang, "The use of a hybrid fuzzy-Delphi-AHP approach to develop global business intelligence for information service firms," *Expert Systems with Applications*, vol. 37, p. 11, 2010.
[26] J. Shin, H. Son, R. Khalil Ur, and G. Heo, "Development of a cyber security risk model using Bayesian networks," *Reliability Engineering & System Safety*, vol. 134, 2015.
[27] J. Zuo, Y. Lu, H. Gao, R. Cao, Z. Guo, and J. Feng, "Comprehensive information security evaluation model based on multi-level decomposition feedback for IoT," *Computers, Materials & Continua*, vol. 65, no. 1, 2020.
[28] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, 2012.
[29] M. Poongodi, M. Hamdi, A. Sharma, M. Ma, and P. K. Singh, "DDoS detection mechanism using trust-based evaluation system in VANET," *IEEE Access*, vol. 7, 2019.
[30] S. H. Almotiri, "Integrated fuzzy based computational mechanism for the selection of effective malicious traffic detection approach," *IEEE Access*, vol. 9, 2021.
[31] M. Petkovic, I. Basicvic, D. Kukolj, and M. Popovic, "Evaluation of takagi-sugeno-kang fuzzy method in entropy-based detection of DDoS attacks," *Computer Science and Information Systems*, vol. 15, no. 1, 2018.
[32] W. H. Lin, P. Wang, B. H. Wu, M. S. Jhou, K. M. Chao, and C. C. Lo, "Behavioral-based network flow analyses for anomaly detection in sequential data using temporal convolutional networks," in *Proceedings of the International Conference on e-Business Engineering*, vol. 41, 2020.
[33] K. Hofer-Schmitz, U. Kleb, and B. Stojanović, "The influences of feature sets on the detection of advanced persistent threats," *Electron*, vol. 10, no. 6, 2021.
[34] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, 2014.
[35] J. Li, K. Cheng, S. Wang et al., "Feature selection: a data perspective," *ACM Computing Surveys*, vol. 50, no. 6, 2017.
[36] T. Abdullah, Y. Bazi, M. M. Al Rahhal, M. L. Mekhalfi, L. Rangarajan, and M. Zuair, "TextRS: deep bidirectional triplet network for matching text to remote sensing images," *Remote Sensing*, vol. 12, no. 3, 2020.
[37] M. H. Abdelbakey and M. S. Shehata, "DP-siam: dynamic policy siamese network for robust object tracking," *IEEE Transactions on Image Processing*, vol. 29, 2020.
[38] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, 2017.
[39] W. Xie, H. Wu, Y. Tian, M. Bai, and L. Shen, "Triplet loss with multistage outlier suppression and class-pair margins for facial expression recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, 2021.
[40] B. Bauer and M. Kohler, "On deep learning as a remedy for the curse of dimensionality in nonparametric regression," *Annals of Statistics*, vol. 47, no. 4, 2019.
[41] J. Li, Z. Tu, B. Yang, M. R. Lyu, and T. Zhang, "Multi-head Attention with Disagreement Regularization, 2020.
[42] J. Shin and J.-H. Lee, Multi-encoder Transformer Network for Automatic Post-Editing, 2019.
[43] R. U. Khan, X. Zhang, and R. Kumar, "Analysis of ResNet and GoogleNet models for malware detection," *J. Comput. Virol. Hacking Tech.*, vol. 15, no. 1, 2019.
[44] J. Li, C. Gu, F. Wei, X. Zhang, X. Hu, and J. Guo, "LightSEEEN: real-time unknown traffic discovery via lightweight siamese networks," *Security and Communication Networks*, vol. 2021, Article ID 8267298, 12 pages, 2021.
[45] S. Kiliçarslan and M. Celik, "RisgELU: a nonlinear activation function for deep neural networks," *Expert Systems with Applications*, vol. 174, 2021.
[46] Uci, “KDD99 data set,” 1999, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[47] H. G. Kayack, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets," in *Proceedings of the 3rd Annual Conference on Privacy, Security and Trust, 2005.
[48] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set, 2009.
[49] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy, January, 2018.
[50] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data, 2019.
[51] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proceedings of the International Carnahan Conference on Security Technology, 2019-October.
[52] M. L. McHugh, "The Chi-square test of independence," *Biochemistry Medica*, vol. 23, no. 2, 2012.
[53] W. R. Rudnicki, M. Wrzesień, and W. Paja, "All relevant feature selection methods and applications," *Stud. Comput. Intell.*, vol. 584, 2015.
[54] D. Kurniabudi, D. Stiawan, M. Y. Bin Bin Idris, A. M. Bambdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection, 2019.
[55] T. D. K. Thara, P. S. Prema, and F. Xiong, "Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques," *Pattern Recognition Letters*, vol. 128, 2019.
[56] A. Nicolson and K. K. Paliwal, "Deep learning for minimum mean-square error approaches to speech enhancement," *Speech Commun*, vol. 111, 2019.
[57] M. Shyamala Devi, R. M. Mathew, and R. Suguna, "Regressor fitting of feature importance for customer segment prediction with ensemble learning using machine learning," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, 2019.