


Article

A BiLSTM-Based DDoS Attack Detection Method for Edge Computing

Yiying Zhang ¹, Yiyang Liu ^{1,*}, Xiaoyan Guo ², Zhu Liu ³, Xiankun Zhang ¹ and Kun Liang ¹ ¹ College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China² Information and Communication Company, State Grid Tianjin Electric Power Company, Tianjin 300140, China³ State Grid Information and Communication Industry Group Co., Ltd., Beijing 100070, China

* Correspondence: liuyy0625@163.com

Abstract: With the rapid development of smart grids, the number of various types of power IoT terminal devices has grown by leaps and bounds. An attack on either of the difficult-to-protect end devices or any node in a large and complex network can put the grid at risk. The traffic generated by Distributed Denial of Service (DDoS) attacks is characterised by short bursts of time, making it difficult to apply existing centralised detection methods that rely on manual setting of attack characteristics to changing attack scenarios. In this paper, a DDoS attack detection model based on Bidirectional Long Short-Term Memory (BiLSTM) is proposed by constructing an edge detection framework, which achieves bi-directional contextual information extraction of the network environment using the BiLSTM network and automatically learns the temporal characteristics of the attack traffic in the original data traffic. This paper takes the DDoS attack in the power Internet of Things as the research object. Simulation results show that the model outperforms traditional advanced models such as Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) in terms of accuracy, false detection rate, and time delay. It plays an auxiliary role in the security protection of the power Internet of Things and effectively improves the reliability of the power grid.

Keywords: distributed denial of service attacks; attack detection; edge computing; bidirectional long short-term memory; power Internet of Things



Citation: Zhang, Y.; Liu, Y.; Guo, X.; Liu, Z.; Zhang, X.; Liang, K. A BiLSTM-Based DDoS Attack Detection Method for Edge Computing. *Energies* **2022**, *15*, 7882. <https://doi.org/10.3390/en15217882>

Academic Editors: Chun-Yen Chang, Charles Tijus, Teen-Hang Meen and Po-Lei Lee

Received: 27 August 2022

Accepted: 20 October 2022

Published: 24 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the expansion of the power grid, various types of power terminal devices are increasing year by year. A large number of IoT terminal devices lacks security protection devices such as firewalls, so these devices are extremely vulnerable to malicious code control. Moreover, it is difficult to identify and defend against highly concealed and complex grid attacks in the new situation with traditional security protection methods. Distributed Denial of Service (DDoS) attacks are carried out on other important IoT devices by taking control of a malicious IoT device. In February 2022, Ukraine was hit by a massive DDoS attack that led to the shutdown of several military websites, including the Ministry of Defence and the Armed Forces of Ukraine. The frequency of IoT security incidents has made it clear that there is a need to strengthen the protection of power IoT security.

DDoS attacks create massive amounts of attack traffic by forming massive botnets that load servers and network links, thus consuming server resources such that they fail to respond to service requests accordingly. According to the data, most of the detection methods for DDoS attacks usually perform attack detection on servers or cloud centres. However, the server side not only has to perform DDoS attack detection, but it also has to handle requests from all parties, so there are situations such as missing network connection contexts. This allows server-side DDoS attack detection methods to process and analyse only incomplete network traffic packets, which in turn results in DDoS attack anomalous traffic not being immediately distinguishable. In order to solve the above problem, it is

necessary to forward all the network traffic in the same communication cycle to the free network segment for the necessary traffic cleaning, but this process will add more burden to the already blocked network, resulting in an inability to respond in real time to user requests, which affects the user experience. Moreover, most of the current detection methods use machine learning algorithms; however, machine learning algorithms overemphasize feature selection and parameter training. Deep learning methods can effectively solve the problem of classifying huge amounts of data in real web application environments. By using deep learning methods, the application of traditional machine learning methods in attack detection has been dramatically improved. The PC side of the network is complicated by DDoS attacks on servers, mainly from PCs as intermediate jumping-off points. However, for power IoT devices, there are different types of power IoT devices with specific traffic characteristics. Therefore, this paper proposes a method to detect and defend DDoS attacks at the source where they are launched. That is to say, using an edge computing approach, the IoT end devices under its jurisdiction are controlled by edge nodes to communicate.

To address the above limitations, we propose a DDoS attack detection method in the power IoT based on edge computing and Bidirectional Long Short-Term Memory (BiLSTM). The model uses the concept of edge computing to design a distributed detection method based on BiLSTM neural networks. Specifically, network services generated by IoT devices managed by them are detected by edge nodes using a DDoS attack detection model.

The contributions are as follows:

- We characterize network flow characteristics by defining statistical characteristics of IP packets.
- We design a BiLSTM-based network traffic prediction model that is able to learn and determine the relationship between packets of the same data stream as a whole.
- We propose a detection framework based on edge computing to shift the detection of DDoS attacks from the cloud centre to the edge nodes.

The rest of the paper is described below. In Section 2 the relevant work of various experts and scholars is reviewed. Section 3 details the DDoS attack detection method based on edge computing and BiLSTM. Section 4 focuses on the simulation experiments. Section 5 analyses the various scenarios of the simulation experiments, compares them with the latest state-of-the-art, and identifies the limitations of the current technology. Section 6 concludes the paper and the next research outlook.

2. Related Work

DDoS attack detection technology is an attack detection method that has been commonly adopted internationally in recent years. At present, scholars at home and abroad have conducted more in-depth research on this technique, mainly including: statistical-based learning, machine-based learning, and three deep learning-based categories.

Statistical learning-based methods are used to predict whether traffic is legitimate or not through entropy, which can be used to measure information uncertainty. DDoS attack traffic can suddenly increase, unlike the traffic in normal network situations. Therefore, a sudden change in entropy values in general can be analysed as the potential indication of a DDoS attack. A method for generating a flow matrix is proposed in [1] using statistical methods to derive the correlation between normal and abnormal flows. The authors in [2] propose a DDoS attack detection method based on the entropy of the source IP address and cardinality distribution calculation. In [3], a detection model based on a random forest approach is proposed to increase the stability of the fit with a random forest model to build the detection model. An adaptive approach is proposed in [4]; this method is used for analysis in order to find the attacker to detect the attack traffic. The authors in [5] introduce a statistical metric called FFSc, which is able to extract three features of network traffic to analyse the behaviour of network traffic for attack detection. However, when the difference between attack traffic and normal network traffic is not significant, then the method may not be able to accurately distinguish normal traffic from attack traffic. The authors in [6] propose a detection method based on active entropy, which detects traffic by

active entropy and flags the detected abnormal traffic. In [7], flow correlation coefficients are compared to discriminate normal traffic from attack traffic. A kernel-based learning algorithm is proposed in [8] that uses entropy features to distinguish normal traffic from attack traffic. The Holder method based on wavelet analysis is proposed in [9] for DDoS detection. In [10], the amount of data received and the distance of information within a specified time window are used to detect whether an attack has occurred. In [11], a plain Bayesian-based detection method is proposed. In this method, a knowledge base is first built to vectorise the data set, and then the vectors are detected using a streaming algorithm and Bayesian methods to finally determine whether an attack has occurred. In [12], a subordination function is established, which is thresholded to determine whether an attack has occurred in the network. However, there are limitations to feature learning-based detection methods. The type and variation of traffic varies in different attack scenarios, so it is difficult to set a suitable detection threshold.

Machine learning-based attack detection methods described in the modern literature are excellent learning methods used to characterise data to identify patterns in a data set, and the technique can provide decision aids for analysts. A detection method suitable for IoT environments is proposed in the literature [13]. The model is deployed in all nodes of the distributed environment, and the parameter information can be shared between the nodes. In [14], the detection of DDoS in SDN networks is proposed by using SVM to classify the feature vectors. In [15], a detection method that uses multiple classifiers for classification is proposed, which combines the results of all classifiers to make a judgement. A semi-supervised DDoS detection method based on entropy estimates, co-classification, information gain ratio, and extra tree integrated classifier is proposed in the literature [16]. In Reference [17], a cloud source-side DOS attack detection system based on machine learning technology is proposed. A DDoS detection method based on the BM-HG-GSO algorithm and firefly swarm optimization algorithm is proposed in [18]. A method for DDoS attack detection via support vector machines is proposed in [19]. Classifiers are tools that classify data based on specific features or patterns displayed in the data, but often there are distinctly different features from normal and attack traffic, such as average size, transmission rate, and arrival time, and although these features differ significantly between attack and normal transmissions, long-term reliance on feature engineering does not accommodate the high variability of current traffic and does not allow for long-term sequences. It is difficult to detect low-rate DDoS with low accuracy.

Deep learning-based attack detection methods are used to obtain the difference between an attack and normal business through a set of consecutive network packets. The authors in [20] propose the training of models in an SDN environment using each of the three deep learning algorithms. Reference [21] constructs a DDoS attack detection model based on BP neural network. A method that combines features extracted from domain knowledge and combines them with BiLSTM networks for the detection of low-rate DDoS attacks is proposed in [22]. A method that extracts feature values multiple times and combines them with LSTM networks for attack detection is proposed in [23]. A CNN-LSTM3 model is constituted in [24] using a gradient descent algorithm for updating the weights during backpropagation. A unique thermal encoding of the raw traffic data corresponding to the data set and training with the CNN algorithm and CNN-LSTM algorithm, respectively, is proposed in [25]. A recursive deep neural network is designed in [26] to learn patterns from network traffic sequences and track network attack activity, and the core of the method is to convert packet detection to window detection. In reference [27], a DDoS attack detection method based on information entropy and deep learning is proposed. A CNN model is used in [28] to detect attack traffic in real time and at a lightweight level, avoiding threshold setting for statistical detection techniques, reducing the reliance on human information, and enhancing the reliability of the network. A feature extraction algorithm using discrete wavelet transform with variance fractal dimension trajectories is proposed in [29] to maximise the sensitivity of CNNs in detecting DDoS attacks. Although the authors claim that their method can perform real-time detection of DDoS attacks in a

range of environments, there are no performance test reports to support this conclusion. The authors in [30] proposes a new lightweight microservices mobile cloud framework that replaces the heavyweight virtual machine based on mobile cloud computing.

After summarising and analysing the work of researchers, this paper proposes a deep learning-based detection method. The method identifies DDoS patterns by taking into account the temporal correlation of the traffic and encapsulating the learning of features through different types of neural networks. In addition, the BiLSTM neural network can learn the obvious and important features between traffic flows through forward and backward sequences to determine whether the traffic is legitimate or not. At the same time, using edge computing, real-time detection of edge nodes can be achieved, improving detection efficiency.

3. Proposed Method

Distributed Denial of Service (DDoS) is an attack method commonly used by hackers. It has the characteristics of simplicity and efficiency and is the most difficult attack to prevent. It generally refers to a zombie computer composed of one zombie or several zombies sending a large number of invalid data packets or additional business requirements to the attacked smart agricultural system, and this attack is often continuous. The consequence of the attack is that the smart agricultural system is unresponsive, unusable, or completely paralyzed. A schematic diagram of a DDoS attack is shown in Figure 1.

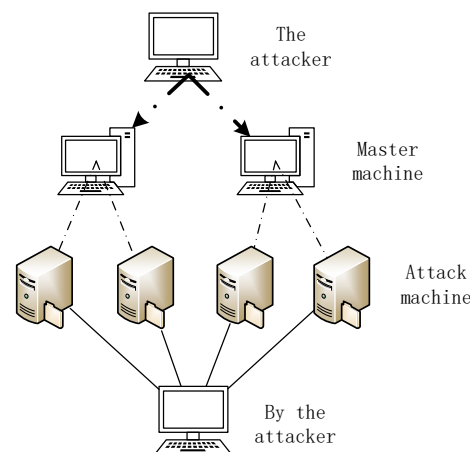


Figure 1. Schematic diagram of a DDoS attack.

3.1. Basic Components of the Electricity Internet of Things

The electricity Internet of Things is an important part of supporting the operation of power production, mainly containing power monitoring systems, distributed power systems, etc. Because of the rapid development of the electricity Internet of Things, the previously closed power system becomes open. There have been many incidents around the world where attacks on the electricity Internet of Things have caused power outages in entire countries. Countries such as Ukraine and Venezuela have experienced massive power outages as a result of cyber-attacks. For the security of the national grid, it is imperative to improve the security of the electricity Internet of Things. The detection of DDoS attacks on the electricity Internet of Things is an important part of this.

The electricity Internet of Things consists of three layers: cloud, management, and end.

Cloud: It consists of a master station. It is responsible for aggregating all kinds of sensor information collected and controlling the execution of many actuators and the visualisation of data in the cloud.

Management: This is the communication layer of the electricity IoT. It mainly includes the 4G/5G network, wired network, etc.

End: This layer is the power end devices of the electricity IoT, such as smart meters as well as charging piles, which sense grid information through sensors.

Figure 2 is a schematic diagram of the cloud-management-end architecture of the electricity Internet of Things.

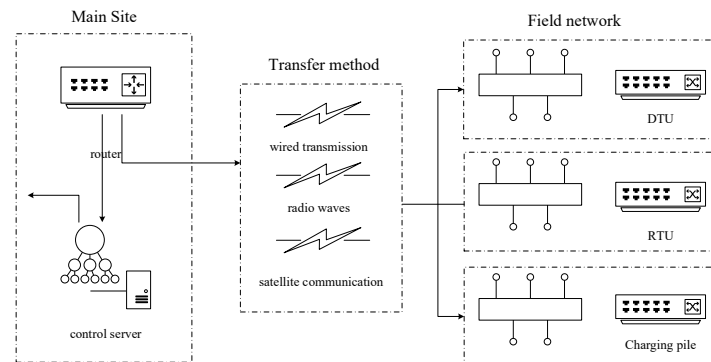


Figure 2. Electricity IoT architecture diagram.

3.2. DDoS Attack Detection Framework Based on Edge Computing

Figure 3 shows the overall frame diagram. This detection method is similar to the previous network intrusion detection system. The edge-side attack detection method proposed in this paper has various functions such as data collection, data pre-processing, and attack detection. Furthermore, the different modules can be extended according to network scale and traffic characteristics.

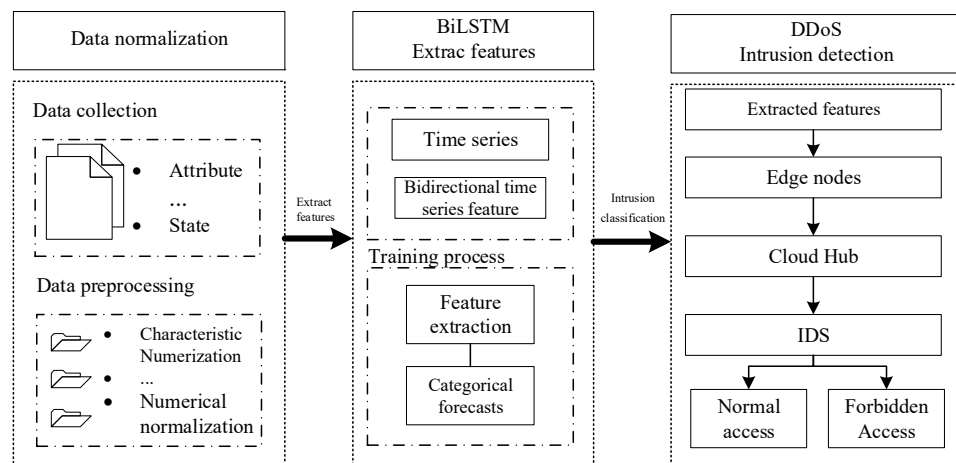


Figure 3. Overall frame diagram.

The main processes of the method include:

- Step 1: The collection and processing of this data are conducted through the data collection module;
- Step 2: The information on the characteristics of the critical services is fed into the detection module for analysis through the feature extraction module;
- Step 3: When a DDoS attack is detected, an attack response is performed.

As shown in Figure 3, when certain terminal devices of the power Internet of Things receive data streams, the data are first pre-processed, using the BiLSTM module to form feature vectors and filter the data traffic. The feature vectors extracted by BiLSTM are collected through edge nodes and uploaded to the cloud centre. If the cloud centre of the feature vector has historical data, the historical data of the cloud centre are directly used for judgement. However, if the extracted feature vectors are not in the historical data, an IDS is used to determine if their device is under attack. Once an anomaly is detected, the traffic is treated as a DDoS attack. At this point, the feature vectors and associated data are updated to the cloud centre, and the updated model is distributed to the edge nodes.

The BiLSTM neural network is used to identify if an attack has occurred through feature extraction and constant updates from the cloud centre.

Figure 4 shows the architecture of the DDoS attack detection network based on edge computing. In the proposed scheme, DDoS attack detection is performed through edge nodes. The edge nodes are used to detect network traffic generated by the IoT devices they manage, identifying normal traffic and DDoS attack anomalous traffic. The network architecture is mainly divided into a terminal layer, an edge layer and a cloud centre layer. The functions of each layer in the network architecture are described in detail next.

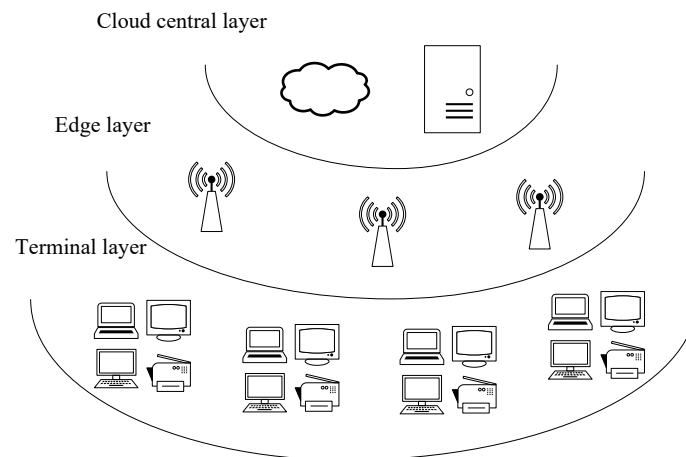


Figure 4. DDoS attack detection network architecture diagram based on edge computing.

Terminal layer: The terminal layer is at the bottom layer of the network architecture and includes all kinds of terminal devices in the power Internet of Things, which can collect various data. DDoS attacks often come from these end devices.

Edge layer: The edge computing layer is composed of numerous edge nodes, located in the middle of the terminal and the cloud centre layer. The edge part of this approach is mainly responsible for DDoS attack detection, DDoS attack detection data collection, and storage. The processed real-time data are uploaded to the cloud centre node, and DDoS attack devices are screened.

Cloud-centric layer: The cloud-centric layer is the highest layer of the entire detection framework, which can train the detection model and perform global regulation. The detection framework collects data at the edge, trains the model at the central node, and distributes the trained model from the central node to the edge nodes for effective edge detection.

3.3. Bidirectional Long Short-Term Memory Network

There are two research tasks in this paper: One is to efficiently and accurately identify the attack flow caused by the DDoS attack and the normal legitimate traffic. Second, during the attack, the traffic has a short burst. Therefore, we analyse the information contained in the data packets of each packet and the characteristics between the packets. Therefore, in the research process of this paper, the bidirectional long-term and short-term memory network are used to perform feature learning on the forward and backward and bidirectional information of the sequence data and then evaluate the value of its output.

Bidirectional Long Short-Term Memory (BiLSTM) consists of two parallel LSTMs: one processes data in a clockwise fashion; the other is counter clockwise. At each moment, the hidden state of BiLSTM is the combination of the two states before and after, which can hide the current and future states.

LSTM generally contains three gates at each sequence time t : forget gate, input gate, and output gate. In this paper, the current input vector x_t , the state memory unit c_{t-1} and the hidden state h_{t-1} of the previous sequence are jointly entered into the forgetting gate. The output f_t of the forget gate is obtained through a sigmoid activation function.

The calculation formula is:

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \quad (1)$$

where W_f , U_f is the weight, and b_f is the bias.

The input gate is divided into two parts: The first part uses the sigmoid activation function, and the output is i_t ; the second part uses the tanh function, and the output is a_t . The two parts together determine the vector that needs to be retained in the state memory unit.

The calculation formula is:

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \quad (2)$$

$$a_t = \tanh(W_a h_{t-1} + U_a x_t + b_a) \quad (3)$$

where W_i , U_i , W_a , U_a is the weight, and b_i , b_a is the bias.

Updating the gate state, C_t consists of two parts. The first part, C_{t-1} , is the product of the output f_t of the forget gate, and the second part is the product of the output i_t and a_t of the input gate.

$$C_t = C_{t-1} \odot f_t + i_t \odot a_t \quad (4)$$

where \odot is the Hadamard product.

The update of the hidden state h_t consists of two parts. The first part is o_t , which is obtained from the previous sequence of hidden states h_{t-1} , sequence data x_t , and the activation function sigmoid. The second part consists of the hidden state c_t and the tanh activation function, which is:

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(C_t) \quad (6)$$

BiLSTM trains the model, as shown in Algorithm 1.

In a bidirectional LSTM network, two LSTMs are fed into one LSTM each. It is different from the LSTM's unidirectional information state to predict the output of the next state; it combines the joint embedding sequence (forward) and the corresponding reverse embedding sequence (reverse) of the two LSTMs and combines the information of the two LSTMs. This makes it possible to predict the subsequent temporal output, which is the output of the last two LSTMs. The BiLSTM structure is shown in Figure 5.

3.4. Temporal Feature Extraction Model

In this model, some IP datagrams are arranged into a forward sequence, and then the forward sequence is fed into the forward LSTM network, and the reverse sequence is fed into the backward LSTM network. Next, the sub-streams in the session stream are analysed bidirectionally for the time-series feature relationships between them using supervised learning from the LSTM layers in both directions. Finally, all datagrams of the whole data stream are judged to be attack traffic or not, and the results of this model are output to the fully connected layer.

After feature extraction in the BiLSTM layer, the assignment of predictive labels to each sub-stream needs to be achieved through the dropout layer and the fully connected layer. During training, the dropout layer is used to avoid the phenomenon of network overfitting. Therefore, the connections between the BiLSTM layer and the fully connected layer are randomly removed during each iteration. In the testing phase, the dropout layer is removed from the neural network, and all the connections are retained in order to facilitate the classification prediction. In particular, the dropout layer can be effective in improving the performance of the network model when the amount of data is small. The fully connected layer is used to flatten the output of the BiLSTM into a more meaningful

vector of feature dimensions, which is then fed into the output layer using SoftMax for the final classification prediction. The temporal feature extraction model is shown in Figure 6.

Algorithm 1 BiLSTM Training

Input: D : query sample set $x \in \mathbb{R}^D$

Output: (y) : the predicted label

1 **At the edge node:**

2 Collect Data

3 Process Data

4 Upload Data to the cloud central node

5 **At the cloud central node:**

6 Begin:

7 **for** Data Processed in Training and Test Sets:

8 1 Extract Features(x)

9 2 Extract Labels(y)

10 **for** Features in x

11 1 Encode Features

12 **for** i in range $(0, n)$:

13 Load BiLSTM Model

14 Fit model

15 Validate model

16 Test model on test sets

17 The trained model is sent to the edge node

18 Return model

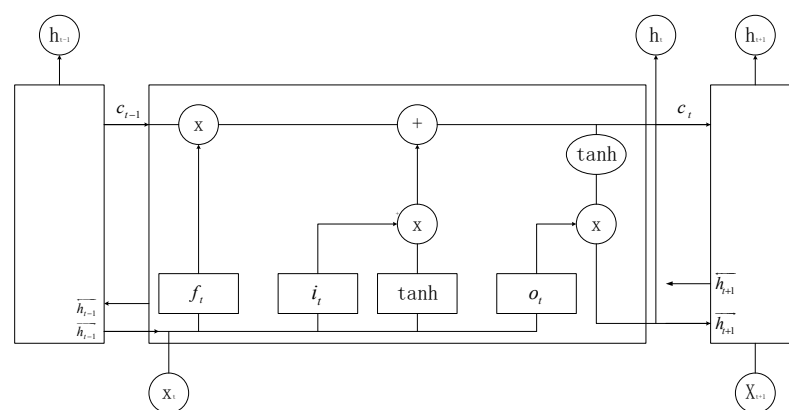


Figure 5. Schematic diagram of BiLSTM structure.

This paper characterises the changing state of network traffic by using the IP packet statistics feature, which samples the network flow at 1 min intervals and accumulates the number of packets in that time period. In this paper, the number of packets per unit time in a normal network is counted on the basis of network traffic. Secondly, the determination of DDoS attacks is conducted, that is to say, the determination of whether there is any abnormality in the statistical characteristic value of IP packets. When an attacker launches

a DDoS attack, abnormal attack traffic is generated per unit time, resulting in a dramatic increase in the number of packets. Therefore, it is possible to determine whether an attack has occurred by determining whether the IP packet statistics feature is abnormal. When there is an abnormal value of IP packet statistics, it can be considered as a DDoS attack. However, this statistical feature only reflects the current changes in network traffic and cannot predict future attacks. It is important to choose the right threshold value for a given data set. If the threshold chosen is too high, it will lead to missed DDoS attacks; if it is too low, there is a risk of false DDoS attacks.

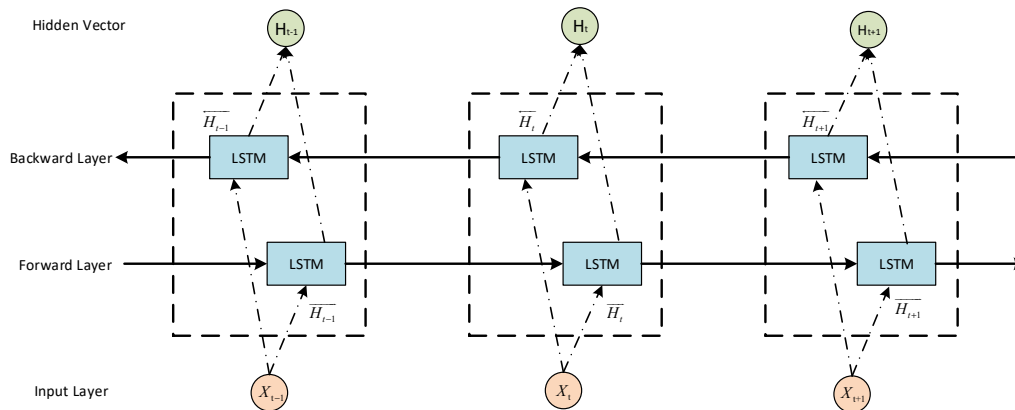


Figure 6. Temporal feature extraction model.

Firstly, through the analysis of the statistical characteristic value intervals of IP data packets of normal traffic and predicted traffic, two statistical characteristic value intervals of IP data packets of normal traffic and predicted traffic are obtained:

$$NOR = [nor_1, nor_2 \dots nor_n] \quad (7)$$

$$PRE = [pre_1, pre_2 \dots pre_n] \quad (8)$$

From this, the maximum value MAX of the statistical characteristic value interval of the IP data packet of the predicted traffic can be obtained.

Secondly, the draw error AVE of the two eigenvalue intervals can be obtained by calculation:

$$AVE = f(NOR, PRE) \quad (9)$$

The threshold is then set to $V = AVE + MAX$. Assuming that the deviation between the actual data and the predicted result is greater than the threshold R, it can be judged that there is an abnormality in the data flow, and thus the existence of a DDoS attack. The processing flow of its attack detection method is shown in Figure 7.

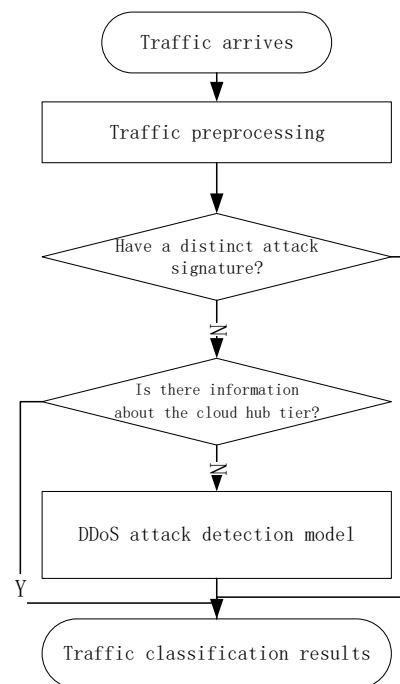


Figure 7. Flowchart of attack detection method.

4. Simulation Experiment

The basic idea of a DDoS attack is to send as many malicious packets as possible to the target server using as many bot hosts as possible. According to the characteristics of a DDoS attack and combined with the power IoT scenario, in our attack scenario, we took the power IoT devices as the constituents of the botnet, and the DDoS attacker invaded a large number of power IoT end devices and used the infected power IoT devices to form a large-scale botnet to send a large amount of attack traffic to the target server in order to exhaust the target server system resources and network resources, thus causing the attacked hosts to be unable to respond to normal service requests. Firstly, in this paper, considering the computing capability of edge computing, IEEE14 node systems were selected as the test environment to allow them to operate normally in an environment without attacks and to collect the data at this point, which was used to distinguish the data generated by DDoS attacks. Second, an IoT DDoS attack occurred. In the experiments in this paper, identically configured computers were selected to simulate edge detectors.

This experiment was divided into three parts.

- Step1: Feature extraction of the raw data. The data were extracted at 1 min intervals, and the number of packets per minute was counted as data features.
- Step2: Prediction using the BiLSTM algorithm. According to the feature extraction method, the training sample, test sample, and attack sample data were sampled separately, and the IP packet statistical feature values were calculated.
- Step3: Performance of DDoS attack detection.

5. Experimental Results and Analysis

In this paper, the accuracy and false alarm rate were selected as the evaluation indicators of this experiment. Accuracy (ACC) refers to the percentage of correctly classified samples, and false positive rate (FPR) refers to the percentage of samples that are incorrectly classified as DDoS.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

$$FPR = \frac{FP}{FP + TN} \quad (11)$$

A confusion matrix is a well-interpreted matrix used in machine learning for summarizing classification predictions, as shown in Table 1.

Table 1. Confusion matrix.

	Predicted Value = 1	Predicted Value = 0
True value = 1	True positive TP	False negative FN
True value = 0	False positive FP	True negative TN

5.1. Analysis of Simulation Experiment Results

The experimental results of the DDoS simulation attack are shown in Figure 8. As the simulation process time progressed, the number of IP packets was small under normal circumstances, the value fluctuated less up and down, and the change of IP packet statistical characteristics was relatively stable; during the attack time period, the number of IP packets increased sharply, and the value fluctuated more up and down. On average, an attack could be verified in 0.0125 s. Assuming that the threshold value in this paper was 7500, at 120 min, the deviation between the real value and the predicted value was too large and exceeded the threshold value R . It could be determined that a DDoS attack occurred.

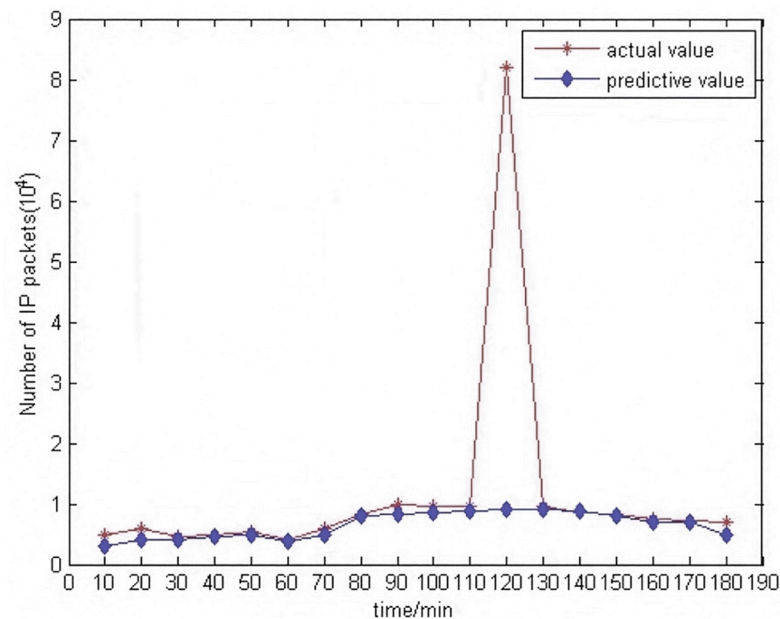


Figure 8. Results of the first DDoS simulated attack experiment.

In the second simulated attack, the threshold R was still assumed to be 7500. As shown in Figure 9 below, as time passed, it could be seen that at 30 min, 60 min, and 160 min, although the deviation between the true value and the predicted value was large, it did not exceed the threshold and was not judged as a DDoS attack. At 80 min and 120 min, the deviation between the true value and the predicted value was very large, exceeding the threshold R . Then it could be accurately determined that a DDoS attack occurred. It could be seen that the detection model was able to detect DDoS attacks where the deviation between the true value and predicted values exceeded the threshold.

5.2. Model Performance Analysis

As shown in Figure 10, the DDoS attack detection rate increased as time increased, while the false detection rate did the opposite. This is because the time had just started, the time to learn the attack features was short, and the attack detection was not accurate. As the learning time increased, the detection rate of DDoS attacks eventually stayed around 95.96%, and the false detection rate remained around 4.04%.

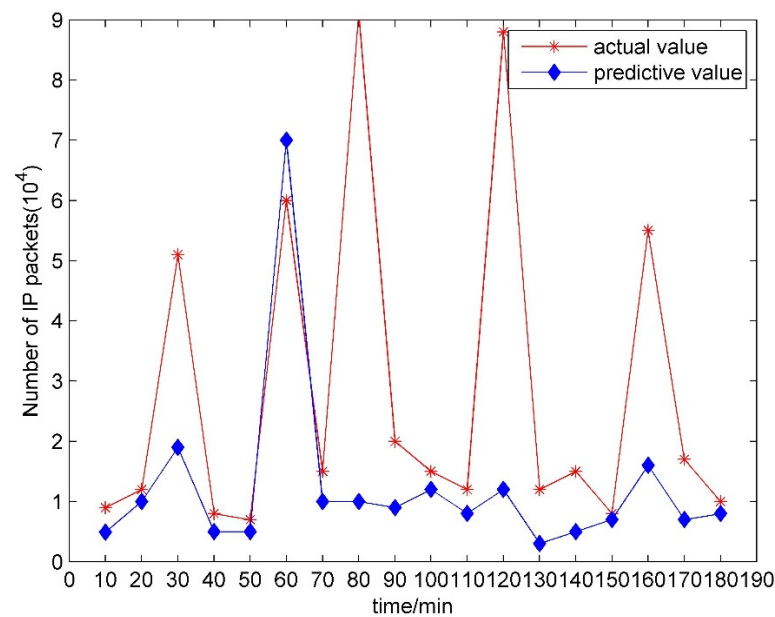


Figure 9. Results of the second DDoS simulated attack experiment.

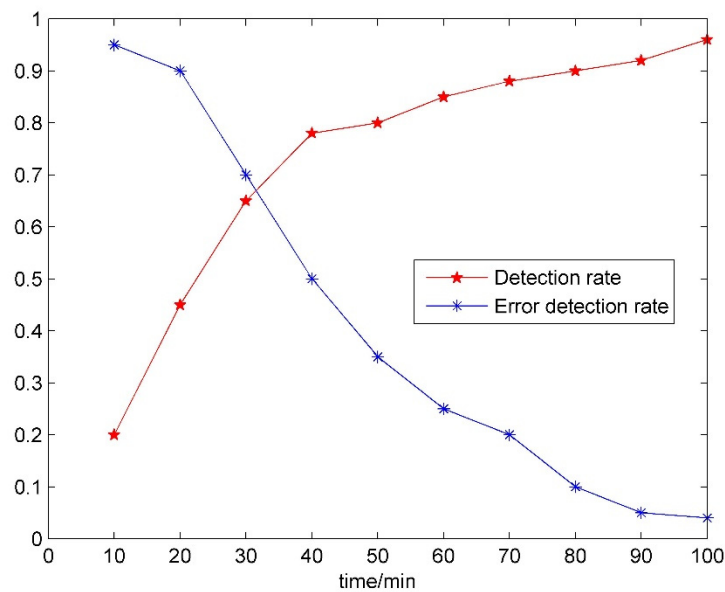


Figure 10. Accuracy and false detection rate.

In this paper, we proposed a detection framework based on edge computing, where the data acquisition module performed data acquisition in close proximity and in real time, and the feature extraction module processed the features extracted from the data simultaneously and handed over the processed data to the central node for learning and distributed the learned detection model to each edge node. The detection module used the trained edge detection nodes to perform detection. When an attack was detected it was fed back to the response module in a timely manner.

The attackers mentioned above took a large number of vulnerable IoT devices and formed a large botnet to launch DDoS attacks. As the number of IoT devices increased, the detection rate of both detection methods for DDoS attacks increased accordingly. The edge computing-based scheme proposed in this paper was compared with an approach that does not use an edge computing scheme.

As can be seen in Figure 11, with and without the edge computing solution, the detection rate of both models increased as the number of IoT devices increased, but when

the devices were larger than 30, the detection rate with the edge-based computing solution was significantly higher than that without the edge computing solution, and the detection rate tended towards 95% steadily, while the detection rate without the edge computing solution was relatively stable in a short period of time. As the number of IoT devices grows, cloud centres will not only have to perform attack detection but also receive requests from all parties at the same time, which will result in very fast packet arrival rates and missing network connection contexts, for example. By introducing the concept of edge computing, DDoS attack detection can be performed at the edge nodes. Using this approach, DDoS attack tasks can be migrated from the cloud centre to the edge nodes, thereby reducing the computational load on the cloud centre. It can be seen that the edge-based computing solution used in this paper is significantly more efficient and applicable compared to the non-adopted edge computing solution.

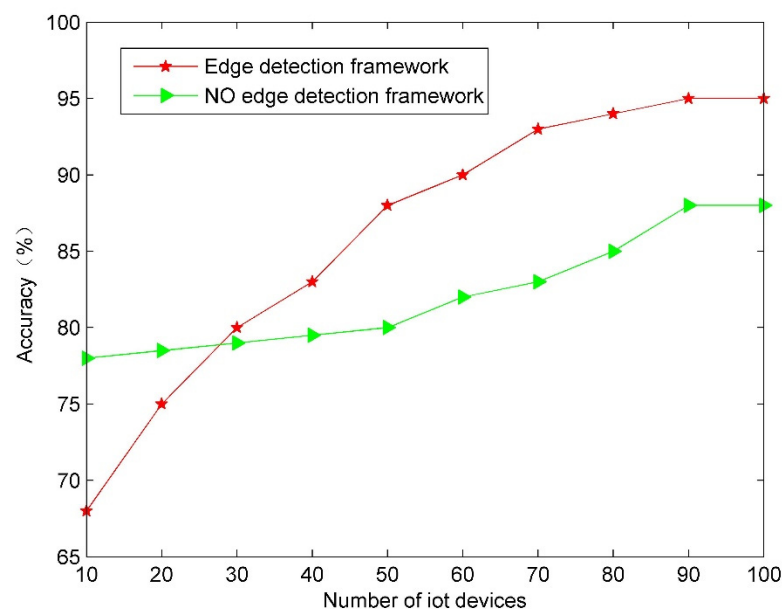


Figure 11. Comparison of detection rates.

Three sets of comparison tests were still set up to further demonstrate the applicability of the BiLSTM algorithm proposed in this paper. Since DDoS attack patterns are almost always changing, new attack patterns are constantly being created or even combining different multiple attack methods. Therefore, all three methods chosen in this paper were able to focus on data streams from adjacent contextual environments and also on the impact of long distance streams on the analysis of the results.

The three approaches are very different when it comes to the time series processing problem. Recurrent Neural Network (RNN) can handle some short-term dependence but not long-term dependence. RNN also suffer from gradient disappearance and gradient explosion problems traditionally. LSTM uses only prior unilateral information to value the output resulting in mediocre model performance. BiLSTM is able to learn and determine the relationship between packets of the same data stream as a whole, which means that simultaneous bidirectional scanning can produce more accurate features. As can be seen from Figure 12, by setting the same normal flow and attack streams, the method proposed in this paper had a higher accuracy when compared with the other two methods using BiLSTM.

To further demonstrate the effectiveness of the edge computing scheme proposed in this paper, the edge computing detection method proposed in this paper was compared with the detection method without using edge computing by setting up the same normal and attack streams. Based on the edge computing detection method, edge nodes perform data acquisition in near real-time and process the data simultaneously to extract features.

The cloud computing centre distributes the trained detection model to the edge nodes to implement edge-side detection.

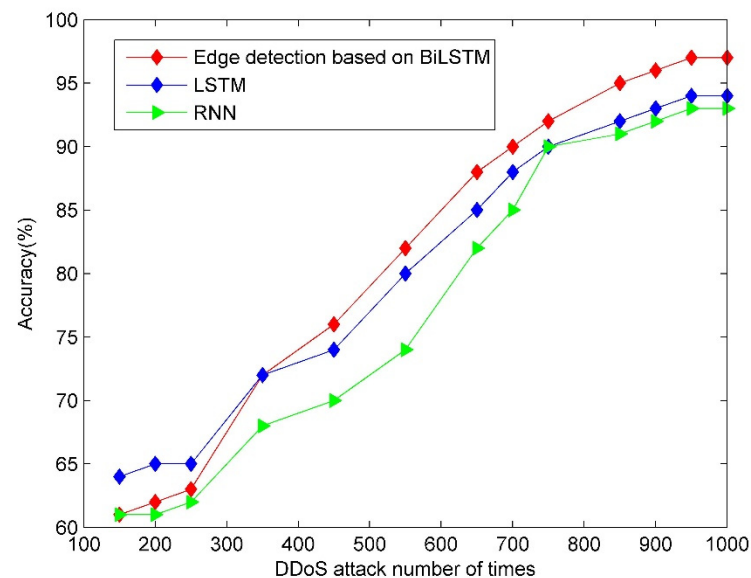


Figure 12. Comparison of the accuracy of different experimental schemes.

As can be seen in Figure 13, with and without the edge computing approach, a DDoS attack occurred at 120 min. The method proposed in this paper could make a quick response and detect the attack effectively. It was able to restore normalcy in a short period of time. In the case without the edge computing approach, when an attack is encountered, it can consume the resources of the attacked host for a long time and is not effectively mitigated for some time.

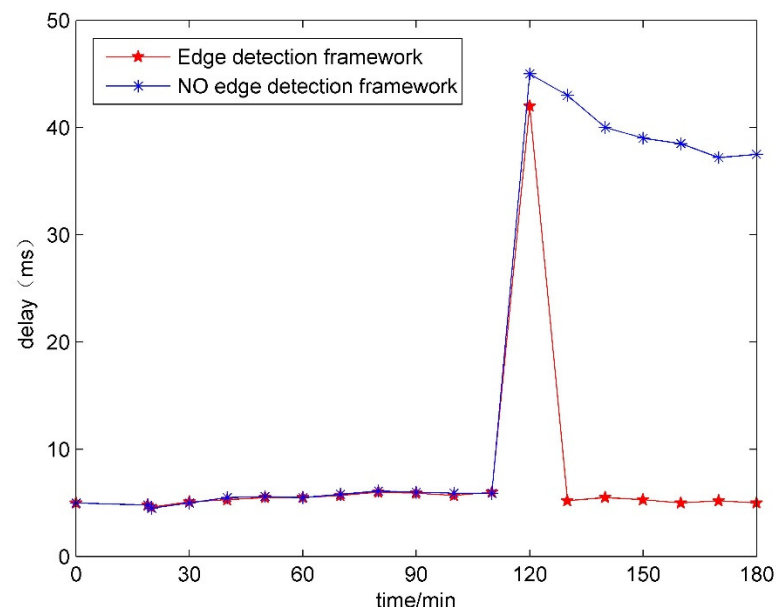


Figure 13. Time delay diagram of different detection frameworks.

In detection methods that do not use edge computing, all data is centralised in the cloud centre for processing, but there is a corresponding communication delay for unified detection through the cloud centre. In the approach using edge computing, the amount of data input in feature extraction and attack detection at each edge node is the traffic data of a portion of the IoT devices currently under jurisdiction. Furthermore, the amount of data

in feature extraction and attack detection at the cloud centre is the data collected by all IoT devices in the smart grid. Compared to the edge nodes, the cloud centre has significantly more data to process and therefore takes more time. In the cloud centre detection method, the data collection of all power IoT devices requires time intervals for different device data collection in order to ensure the normal operation of the devices, so it is more difficult to achieve near real-time detection, and there is a certain detection delay compared to the real-time detection of edge nodes. Therefore, real-time detection via edge nodes effectively reduces the time required for centralised processing. At the same time, the workload of the cloud centre is reduced because the detection is carried out at the edge nodes. The results of the simulation experiments showed that the edge computing scheme used in this paper had higher efficiency compared to that without the edge computing scheme.

5.3. Comparison with Existing State-of-the-Art Technologies

We compared the proposed model with existing state-of-the-art techniques. We compared the performance of the accuracy metric in the most recent state-of-the-art literature separately.

As can be seen from Table 2, the model combining BiLSTM and edge computing proposed in this paper obtained a higher accuracy than other existing state-of-the-art techniques. The accuracy of the model was 95.96%, which is better than other existing models. As can be seen from Table 2, the BiLSTM combined with edge computing model improved the accuracy over the traditional RNN model by 3.36%, the detection accuracy over the traditional LSTM model by 2.21%, and the accuracy over the BiDLSTM model by 1.70%. In addition, higher accuracy was obtained compared to other models. In addition, the BiLSTM model obtained better f-values, making the model superior to existing state-of-the-art methods in detecting attacks.

Table 2. Summary of the proposed model architecture.

Algorithm	Method	Accuracy
AE-LSTM [31] THEODORA [32]	Conventional RNN	92.6
	Conventional LSTM	93.75
	AE + LSTM	90.5
	AE + CNN	92.97
	BiDLSTM	94.26
	Proposed BiLSTM	95.96

5.4. Limitations

Simulation experiments showed that the model is effective in detecting DDoS attacks on the power IoT. However, the area of combining BiLSTM with edge computing needs to be explored in depth. By comparing with other advanced techniques, it can be seen that the proposed BiLSTM combined with edge computing has a more accurate detection rate in identifying DDoS attacks. However, in the complexity and runtime analysis of the combined BiLSTM and edge computing model, it is pointed out that compared with the traditional RNN models and LSTM models, the developed method has higher complexity, requires more training time and delivers the model to each edge node. Although the leakage rate is low, it results in high communication overhead costs. These limitations will be refined in the future.

6. Conclusions

After studying smart grid cybersecurity, it is found that there are still many problems in the existing DDoS attack detection methods, for example, false-positive rate, high false-positive rate, and a relatively single type of detected attack. In this paper, an edge detection framework based on BiLSTM is proposed to detect DDoS attacks in power IoT. This method establishes the BiLSTM neural network prediction model and uses the edge computing method to build an edge detection framework to reduce the delay. By predicting normal network traffic and setting thresholds, it can distinguish abnormal situations caused by

DDoS attacks. This method grasps the changing trends of normal network traffic and can accurately and quickly detect DDoS attacks. Compared with other detection methods, this method has a higher detection rate, lower delay, and lower false-negative rate.

In the simulation experiment, the performance of the attack detection model proposed in this paper is evaluated by simulating the results of the DDoS attack detection rate. The experimental results show that the edge detection model based on BiLSTM can quickly detect malicious attacks, and the detection rate is also high. Compared with the traditional LSTM detection model, the detection model based on BiLSTM and edge computing proposed in this paper is more suitable for the current smart power IoT environment. To sum up, this solution more comprehensively guarantees the security of the power Internet of Things, which has important theoretical and practical significance. Due to the limitations of the experimental environment, it is impossible to test the adaptability of the DDoS attack detection model under massive data. In the future, experiments will be conducted with more power IoT devices, and some state-of-the-art feature selection methods will be explored in combination with the BiLSTM model proposed in this paper.

Author Contributions: All authors contributed to the writing and revisions; writing—review and editing, Y.Z.; writing—original draft, Y.L.; methodology, X.G.; data curation, Z.L.; project administration, X.Z.; supervision, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available upon request due to restrictions of privacy or ethical concerns.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Luo, H.; Hu, G.; Yao, X. DDoS attack detection based on abnormal characteristics of global network traffic. *Comput. Appl.* **2007**, *27*, 314–317.
2. Shi, L.; Zhang, F.; Liu, W. Internet of Things + blockchain helps food quality and safety assurance. *Agric. Technol.* **2019**, *39*, 40–42.
3. Yu, P.; Qi, Y.; Li, Q. DDoS attack detection method based on random forest classification model. *Comput. Appl. Res.* **2017**, *34*, 3068–3072.
4. Zheng, J.; Li, Q.; Gu, G.; Cao, J.; Yau, D.K.; Wu, J. Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1838–1853. [[CrossRef](#)]
5. Hoque, N.; Bhattacharyya, D.K.; Kalita, J.K. A novel measure for low-rate and high-rate DDoS attack detection using multivariate data analysis. In Proceedings of the 2016 8th International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 5–10 January 2016; pp. 1–2.
6. Zhang, M.; Li, Y.; Zhang, P.; Sun, M. A DDoS attack detection method based on Active Entropy under Heavy Traffic. *Appl. Res. Comput.* **2016**, *33*, 2148–2151.
7. Yu, S.; Zhou, W.; Jia, W.; Guo, S.; Xiang, Y.; Tang, F. Discriminating DDoS attacks from flash crowds using flow correlation coefficient. *IEEE Trans. Parallel Distrib. Syst.* **2011**, *23*, 1073–1080. [[CrossRef](#)]
8. Çakmakçı, S.D.; Kemmerich, T.; Ahmed, T.; Baykal, N. Online DDoS attack detection using Mahalanobis distance and Kernel-based learning algorithm. *J. Netw. Comput. Appl.* **2020**, *168*, 102756. [[CrossRef](#)]
9. Ren, Y.; Liu, Y. A DDoS attack detection method based on wavelet analysis. *Comput. Eng. Appl.* **2012**, *48*, 82–88.
10. Behal, S.; Kumar, K.; Sachdeva, M. D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events. *J. Netw. Comput. Appl.* **2018**, *111*, 49–63. [[CrossRef](#)]
11. Durad, M.H.; Cao, Y.; Zhu, L. Two novel trust evaluation algorithms. In Proceedings of the 2006 International Conference on Communications, Circuits and Systems, Singapore, 30 October–1 November 2006; pp. 1641–1646.
12. Yang, J.; Wang, X.; Liu, G. DDoS attack detection method based on traffic and IP entropy characteristics. *Comput. Appl. Res.* **2016**, *33*, 1145–1149.
13. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 761–768. [[CrossRef](#)]
14. Ye, J.; Cheng, X.; Zhu, J.; Feng, L.; Song, L. A DDoS Attack Detection Method Based on SVM in Software Defined Network. *Secur. Commun. Netw.* **2018**, *4*, 9804061. [[CrossRef](#)]
15. Koay, A.; Chen, A.; Welch, I.; Seah, W.K. A new multi classifier system using entropy-based features in DDoS attack detection. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 162–167.

16. Idhammad, M.; Afdel, K.; Belouch, M. Semi-supervised machine learning approach for DDoS detection. *Appl. Intell.* **2018**, *48*, 3193–3208. [[CrossRef](#)]
17. He, Z.; Zhang, T.; Lee, R.B. Machine Learning Based DDoS Attack Detection from Source Side in Cloud. In Proceedings of the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 26–28 June 2017; pp. 114–120.
18. Tan, M. Research and Implementation of DDoS Attack Detection Based on Machine Learning in a Distributed Environment. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2018.
19. Hoyos, L.; Isaza, E.; Vélez, J.; Luis Castillo, O. Distributed Denial of Service (DDoS) Attacks Detection Using Machine Learning Prototype. *Adv. Intell. Syst. Comput.* **2016**, *474*, 33–41.
20. Li, C.; Wu, Y.; Qian, Z.; Sun, Z.; Wang, W. DDoS attack detection and defense based on hybrid deep learning model in SDN. *J. Commun.* **2018**, *39*, 176–187.
21. Miao, X.; Fang, S. DDoS Attack Detection Method Based on ACO-BP Neural Network in SDN. *Data Commun.* **2022**, 42–46.
22. Jiang, W.; Guo, C.; Jiang, C. A low-rate DDoS attack detection method based on BiLSTM. *Comput. Mod.* **2020**, *5*, 120–126.
23. Cheng, J.; Tang, X.; Huang, M.; Luo, Y. DDoS Attack Detection Method and Device Based on LSTM Prediction Model. 201810912851.1, 27 November 2018.
24. Oena, A. A DDoS attack behavior detection method based on deep learning. *arXiv* **2016**, arXiv:1601.04033.
25. Wang, W.; Sheng, Y.; Wang, J.; Zeng, X.; Ye, X.; Huang, Y.; Zhu, M. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* **2017**, *6*, 1792–1806. [[CrossRef](#)]
26. Yuan, X.; Li, C.; Li, X. DeepDefense: Identifying DDoS attack via deep learning. In Proceedings of the 2017 IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, China, 29–31 May 2017; pp. 1–8.
27. Lu, W.; Liu, Y. A DDoS attack detection method based on information entropy and deep learning in SDN. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; Volume 1.
28. Doriguzzi-Corin, R.; Millar, S.; Scott-Hayward, S.; Martinez-del-Rincon, J.; Siracusa, D. LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 876–889. [[CrossRef](#)]
29. Ghanbari, M.; Kinsner, W. Extracting features from both the input and the output of a convolutional neural network to detect distributed denial of service attacks. In Proceedings of the 2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), Berkeley, CA, USA, 16–18 July 2018; pp. 138–144.
30. Sodhro, A.H.; Lakhani, A.; Pirbhulal, S.; Groenli, T.M.; Abie, H. A lightweight security scheme for failure detection in microservices IoT-Edge networks. In *Sensing Technology*; Springer: Cham, Switzerland, 2022; pp. 397–409.
31. Mushtaq, E.; Zameer, A.; Umer, M.; Abbasi, A.A. A two-stage intrusion detection system with auto-encoder and LSTMs. *Appl. Soft Comput.* **2022**, *121*, 108768. [[CrossRef](#)]
32. Andresini, G.; Appice, A.; Caforio, F.; Malerba, D. Improving cyber-threat detection by moving the boundary around the normal samples. In *Machine Intelligence and Big Data Analytics for Cybersecurity Applications Studies in Computational Intelligence*; Maleh, Y., Baddi, Y., Shojaafer, M., Alaza, M., Eds.; Springer: Cham, Switzerland, 2021; pp. 105–127.