Hindawi
Security and Communication Networks
Volume 2021, Article ID 9534016, 10 pages
https://doi.org/10.1155/2021/9534016

*Research Article*

# WEB DDoS Attack Detection Method Based on Semisupervised Learning

**Xiang Yu**,[1] **Wenchao Yu**,[2] **Shudong Li**,[3] **Xianfei Yang**,[1] **Ying Chen**,[1] **and Hui Lu**[3]

[1]*School of Electronics and Information Engineering, Taizhou University, Taizhou 318000, China*
[2]*Computer Science and Technology College, Harbin Engineering University, Harbin 150001, China*
[3]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Shudong Li; lishudong@gzhu.edu.cn and Hui Lu; luhui@gzhu.edu.cn

Received 14 August 2021; Accepted 15 October 2021; Published 29 November 2021

Academic Editor: Zhili Zhou



Since the services on the Internet are becoming increasingly abundant, all walks of life are inextricably linked with the Internet. Simultaneously, the Internet's WEB attacks have never stopped. Relative to other common WEB attacks, WEB DDoS (distributed denial of service) will cause serious damage to the availability of the target network or system resources in a short period of time. At present, most researches are centered around machine learning-related DDoS attack detection algorithms. According to previous studies, unsupervised methods generally have a high false positive rate, while supervisory methods cannot handle large amount of network traffic data, and the performance is often limited by noise and irrelevant data. Therefore, this paper proposes a semisupervised learning detection model combining spectral clustering and random forest to detect the DDoS attack of the WEB application layer and compares it with other existing detection schemes to verify the semisupervised learning model proposed in this paper. While ensuring a low false positive rate, there is a certain improvement in the detection rate, which is more suitable for the WEB application layer DDoS attack detection.

## 1. Introduction

In the era of the prevailing development of the Internet, with the rapid development of the Internet, the services on the Internet are increasing, and all walks of life are inextricably linked with the Internet. Under this trend, people have become increasingly dependent on the Internet; whether it is online shopping or travel, it is closely related to the Internet. However, while the Internet is developing comprehensively and rapidly, the attacks on the Internet continue to exist and change constantly. Among them, WEB applications have become the focus of attacks because of their wide range of uses. Common WEB attacks [1] include WEB DDoS attacks, cross-site scripting attacks, and request forgery attacks. With the development of distributed and the proliferation of botnets, WEB DDoS attacks have become the most threatening attack, which can seriously damage the availability of target networks or system resources during the duration of a short attack.

WEB DDoS attacks have three characteristics: distributed, rapid development, and destructiveness [2]. However, traditional attack detection methods cannot effectively and accurately detect WEB DDoS, and with the development of machine learning, many researchers have used it to detect WEB DDoS attacks. In the machine learning [3–7] algorithm, there are two types: unsupervised learning and supervised learning. However, the unsupervised method alone has a high false positive rate, while the supervised method alone cannot handle a large number of unknown attacks. For the new type of attack of network traffic data, researchers have used *K*-means + C4.5 for attack detection, which has been experimentally proved to have a higher detection rate than the use of supervised or unsupervised algorithms alone, but because of its use of K-means compared with other current machine learning algorithms, the C4.5 algorithm has insufficient performance, so its detection accuracy and false positive rate have a lot of room for improvement. Therefore, this paper will

study and propose a detection method for WEB DDoS attacks.

## 2. Related Work

The focus of this paper is on DDoS attacks in the WEB application layer. Research on this direction has never stopped at home and abroad. Moreover, with the development of machine learning technology, machine learning methods have become a mainstream method in DDoS detection research. Both Kim et al. [4] use machine learning methods to identify network traffic. The former finally derives DBSCAN. It is more suitable for clustering. The latter shows that the support vector machine (SVM) performs better in detecting attacks. Calix and Rajesh [5] tested the SVM algorithm on the NSL-KDD data set. The accuracy rate is less than 80%. Literature [8] clusters users by the *K*-means clustering algorithm, which can be achieved by uniform clustering. Panda [9] compared several classification algorithms, in which a random forest-based set classifier can achieve 99% accuracy. Muniyandi et al. [7] proposed a hybrid algorithm using *K*-means + C4.5 for attack detection whose detection rate is higher than the one using a supervised algorithm or an unsupervised algorithm alone.

The DDoS detection methods in the literature are mainly divided into two categories: unsupervised methods and supervised methods. There are two main problems depending on the benchmark data set used:

(1) The false positive rate of unsupervised methods is often high.

(2) The supervisory method cannot handle large or new types of attack network traffic data, and its performance is often limited by noise and irrelevant data.

(3) Since the *K*-means + C4.5 method uses the *K*-means and C4.5 algorithms, its performance is insufficient when compared with other current machine learning algorithms, so its detection accuracy and false positive rate has a lot of room for improvement.

Based on the above three problems, this paper proposes a semisupervised learning model combining spectral clustering and random forest to detect WEB DDoS. Compared with the existing scheme, it has a high performance rate and low false positive rate performance improvement, which is more suitable for current WEB DDoS attack detection.

## 3. Detection Methodology

In this paper, the semisupervised learning [10–15] model combined with unsupervised learning and supervised learning methods is used to detect WEB DDoS attacks, and the choice of learning methods has a great impact on the performance of this model.

First, for the unsupervised model [16–22], it includes DBSCAN, *K*-means, and spectral clustering. The DBSCAN algorithm [23] has a long convergence time when the sample data is too large and is not suitable for the big data network environment. Compared with *K*-means, the spectral clustering algorithm is very effective for the clustering of sparse data, while *K*-means is difficult to do. In addition, spectral clustering is processing the network traffic data because of the dimensionality reduction processing. In high-dimensional data, the complexity is lower than traditional clustering methods such as *K*-means. Therefore, this paper chooses spectral clustering as an unsupervised learning algorithm for semisupervised learning models.

Second, for the supervised model [24, 25], the most commonly used algorithms include SVM, Naive Bayes, C4.5, and Random Forest. Lee et al. [26] compared the above classification algorithm, which proved that the random forest is the best classification effect among these algorithms. Panda et al. [6] also compared several supervised algorithms with two types of classifications. The cluster classifier based on random forest is optimal and can achieve 99% accuracy. Based on the above research, this paper chooses random forest as the supervised learning algorithm of semi-supervised learning model.

This section applies the semisupervised learning model based on spectral clustering algorithm and random forest combination to detect WEB DDoS attacks. Firstly, the principle and characteristics of spectral clustering in the model are introduced, and then the classification algorithm applied to the model is random forest. The principle and advantages are introduced. Finally, the design of WEB DDoS detection model framework based on semisupervised learning combined with spectral clustering and random forest is introduced.

*3.1. Spectral Clustering Algorithm Model.* The clustering algorithm used in this paper is spectral clustering, and the spectral clustering algorithm is theoretically used to establish spectra. Compared with the traditional clustering algorithm, spectral clustering can better divide the sample data into clusters with high similarity regardless of the sample space. The principle of the spectral clustering algorithm [27] is as follows. Firstly, the data of the sample data set is transformed into a similar matrix that reflects the similarity between the sample data. Next, the matrix eigenvalues and eigenvectors are solved. Finally, select the feature vector that can cluster the data relatively well. This algorithm can converge to the global optimal solution. At the beginning of spectrum clustering, there are few studies on computer applications. The field of powerful clustering ability is computer vision and VLSI design. At present, machine learning is also applied to solve clustering problems and research at home and abroad. The efforts of scholars have become a hot clustering algorithm.

The spectral clustering algorithm is divided into two types according to different division criteria: 2-way and k-way. The 2-way method includes PF algorithm, SM algorithm, and Mcut algorithm. The previous spectral clustering algorithm generally uses the 2-way method to divide and cluster data samples. However, in most of the current research, it is found that the result of dividing and clustering by more feature vectors and using k-way method is better. Ng et al. [28] proposed the NJW algorithm based on k-way method by solving the first $k$ largest eigenvalues of the Lagrangian matrix and its corresponding eigenvectors and

orthogonalizing the $k$ eigenvectors. The sample space $R_k$ is obtained, so that the original data and each data point in the $R_k$ space form a one-to-one representation, and finally clustering is performed in the $R_k$ space.

The general process of the spectral clustering algorithm based on the NJW algorithm is shown in Figure 1.

Among them, when constructing the Laplacian matrix, memory consumption can be saved by writing the operation result to the disk, and when the row vector of the feature vector matrix is converted into a unit vector, it is calculated by

$$Y_{ij} = \frac{X_{ij}}{\sqrt{\sum_j x_{ij}^2}}. \quad (1)$$

When the spectral clustering is finally clustered by $K$-means, it is necessary to satisfy the condition that the data sample $y_i$ is divided into cluster $j$ if and only if the $i$ row of $Y$ is divided into clusters $j$.

*3.2. Random Forest Algorithm Model.* The random forest [29] is based on the basic idea of bagging to train a series of decision trees and improve them according to the characteristics of the decision tree. In the random forest training process, it adopts random attribute selection to improve the relative independence of the constructed decision tree to improve performance. Assuming that the number of nodes is $n$, the way in which the traditional decision tree selects the best attribute is based on all the attributes of the $n$ nodes, and each node of the decision tree in the random forest is based on $k$ attributes that are randomly selected in advance. The magnitude of the $k$ value is decisive for the degree of randomness and is usually set to $\log_2 d$. In addition, the $k$ value can also be 1 or $d$, which, respectively, represents a random selection of an attribute and a selection method using a conventional decision tree. The specific flow of the random forest algorithm is shown in Algorithm 1.

It can be seen from the training process of random forests that it only makes some minor changes to bagging, adding the randomness of feature attributes on the basis of random samples, and the generalization of the final integration of random forests. The degree of increase is better. Because the random forest algorithm has the advantages of small computational complexity and small difficulty in solving classification problems and often exhibits strong performance in practical applications, this paper also uses random forest as the classifier in the model.

*3.3. Attack Detection Model Framework Based on Semisupervised Learning.* The detection model proposed in this paper is based on the semisupervised learning model. The spectral clustering clustering algorithm introduced in Section 3.1 is used as the unsupervised learning algorithm in the model [30–39]. The abovementioned random forest algorithm is used as the model. There is a supervised learning algorithm. Through the cooperation of these two algorithms, this paper will construct a WEB DDoS attack detection framework based on semisupervised learning. The basic framework and process design are as follows. Since this

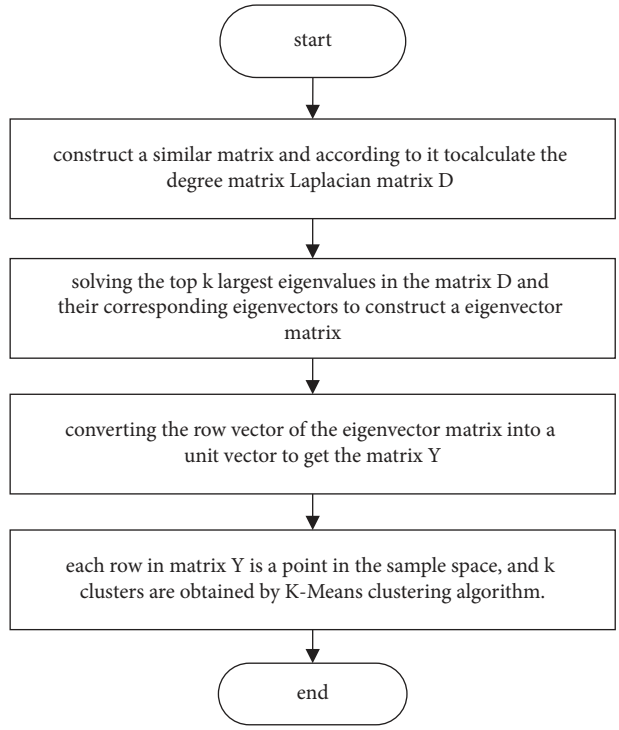


Figure 1: Process of spectral clustering algorithm based on NJW algorithm.

semisupervised learning type detection framework is based on machine learning algorithms, it is similar to the traditional machine learning algorithm [40–45], including the training process and the detection process, and the approximate processing of these two processes is shown in Figure 2.

For the training phase, the defined dataset $S$ is $(X_i, Y_i)$, $i = 1, 2, \ldots, N$, where $X_i$ represents an *N-dimensional* matrix, $Y_i = \{0,1\}$, where 0 represents normal flow and 1 represents abnormal flow. In the training process, the training data set is first divided into $k$ disjoint clusters by spectral clustering. The random forest corresponding to each cluster is then trained with the data in each cluster.

For the detection phase, the spectral clustering method is used to calculate which cluster of the $k$ clusters the test data sample belongs to, and the corresponding random forest classifier is found according to the cluster of the sample data to determine whether the data sample is normal data or abnormal data.

## 4. Experiments

*4.1. Experimental Environment.* Table 1 lists the hardware and software environments used in this experiment.

*4.2. Experimental Program*

*4.2.1. Extraction of Data Set.* This experiment uses the five-fold cross-validation method to test, extract 50,000 data from the NSL-KDD data set, and divide it into 5 equal parts. Each subdata set is divided into four types according to the

(1) Input: training set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
(2) Learning algorithm $A$
(3) Training argument $m$
(4) Output: strong classifier $f(x)$
(5) **begin**
(6) **for** $t = 1, 2, \ldots, T$ **do**
(7) Produced bootstrap samples set and named $S_t$
(8) Train a decision tree $T_j$ on $S_t$
(9) **while** the number of samples corresponding to the leaf node is greater than $n_{\min}$ **do**
(10) Randomly select $k$ variables from all optional $d$ variables
(11) Select from these $k$ variables the variables that can lead to the optimal partition
(12) Divide the node into two subnodes according to the best variable selected above
(13) **end**
(14) **end**
(15) Aggregate $m$ decision trees
(16) **end**

Algorithm 1: Random forest algorithm.

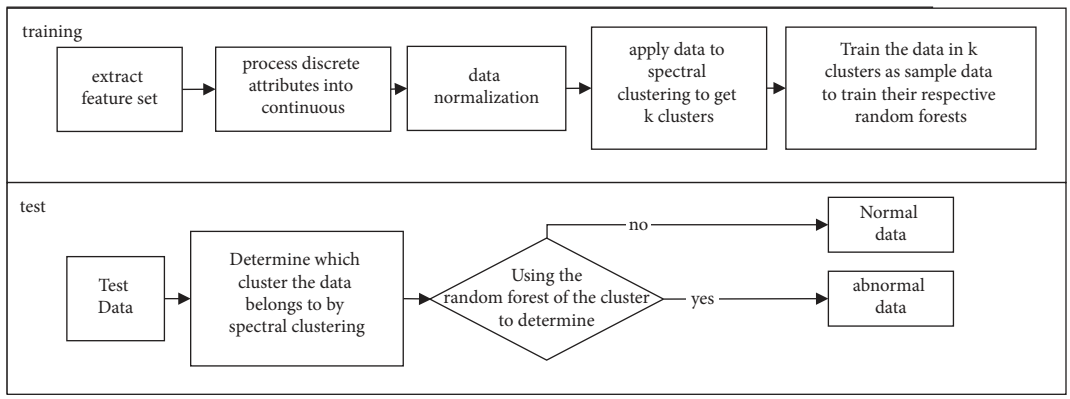


Figure 2: Semisupervised learning model training and testing process.

Table 1: Experimental environment parameters.

| CPU | Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40 GHz |
|---|---|
| Operating system | CentOS 7.0 |
| RAM | 8 GB |
| Programming language | python |
| Template library | sklearn |

upper service type, including HTTP, SMTP, FTP, and others. The type of data in each category contains 40% of the attack data. The details of the data contained in each subdataset are shown in Table 2.

According to the k-fold cross-validation principle, each experiment will select the subset of data from the previous experiment that was not selected in the previous experiment. This model is used to test the trained model, and the remaining word data sets are available. Model training is used for learning, and $k$ experiments are performed in this selection. The experimental results, that is, the performance of the model, are reflected by the average of $k$ experiments. The principle flow of the 50% algorithm and the data set of this experiment are shown in Figures 3 and 4, respectively.

*4.2.2. Data Preprocessing.* The learning model's evaluation rules are learned through the marked connections in the dataset. These connections are TCP data messages sent and received by the same IP address in a unit of time. The connection is marked as normal or abnormal. The features of each dimension of the NSL-KDD data set are divided into discrete and continuous types, and their respective ranges of values are different. Therefore, preprocessing is required for these features. The preprocessing includes continuous discrete feature variables and data normalization. The two processes are described as follows.

First, the discrete feature variables need to be continuous. The NSL-KDD data set contains continuous and discrete variables, and the discrete feature variables cannot be quantized, so the data is applied to the model. Previously, it was to be continuously processed. According to statistics, NSL-KDD contains 7 discrete feature variables, 5 of which can be represented by 0 or 1 values, namely, _guest_login, logged_in, land, flag, and is_host_login feature variables. The service and protocol_type characteristic variables require special conversion because they have several different values. The specific conversion methods are shown in Tables 3 and 4.

Table 2: Subset data type distribution details.

| Service type | Total number of records | Total number of attack records | Attack record ratio (%) |
|---|---|---|---|
| HTTP | 4000 | 1600 | 40 |
| FTP | 2000 | 800 | 40 |
| SMTP | 2000 | 800 | 40 |
| Others | 2000 | 800 | 40 |
| Total | 10000 | 4000 | 40 |

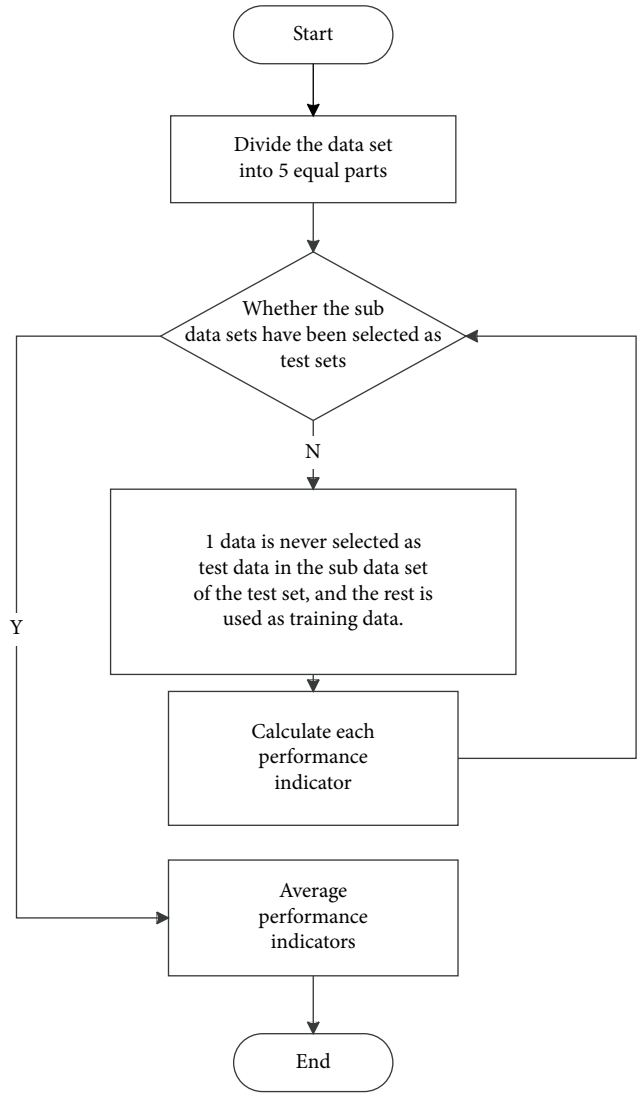


Figure 3: Flow chart of the five-fold algorithm.

Since the classification of data samples in this paper is obtained by calculating the degree of similarity between data samples, through the previous research on data sets, it contains many feature attributes, and the range and unit of each feature attribute are different. In order for the degree of similarity of the calculations to better represent the differences between the samples, data normalization is required. Data normalization refers to scaling feature attribute data proportionally so that the range of values of the data is reduced to a specific interval, i.e., [−1, 1] or [0, 1]. This experiment uses the $z$-score method to normalize the experimental data.

*4.2.3. Performance Criteria.* The performance indicators used to evaluate the experimental results are calculated based on the standard confusion matrix. For the sample data of this experiment, the confusion matrix is shown in Table 5. True positive (TP) refers to a record that is correctly classified as attack traffic, while false positive (FP) refers to a record that is misclassified as attack traffic, true negative (TN) is a record that is correctly classified as normal traffic, and false negative (FN) is a record that is misclassified as normal traffic. The formulas for the performance indicators used are defined as follows:

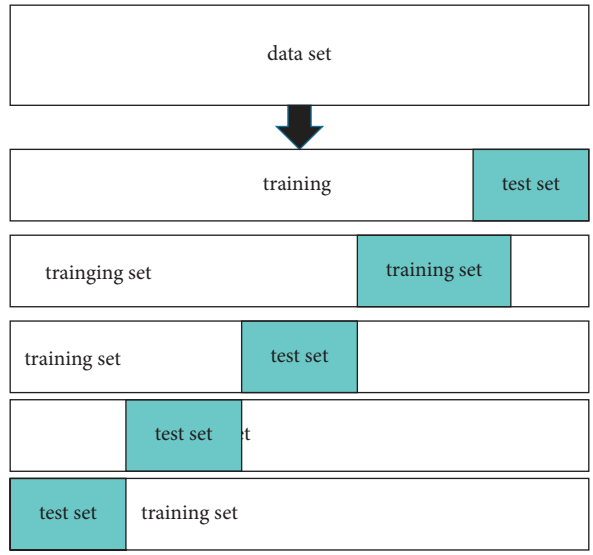


Figure 4: Data set and test set partitioning.

Table 3: Service feature variable transformation.

| Service | Service 0 | Service 1 | Service 2 | Service 3 |
|---|---|---|---|---|
| HTTP | 0 | 0 | 0 | 1 |
| SMTP | 0 | 0 | 1 | 0 |
| FTP | 0 | 1 | 0 | 0 |
| Others | 1 | 0 | 0 | 0 |

Table 4: Protocol type feature variable conversion.

| Protocol type | Protocol type 1 | Protocol type 2 | Protocol type 3 |
|---|---|---|---|
| TCP | 0 | 0 | 1 |
| UDP | 0 | 1 | 0 |
| ICMP | 1 | 0 | 0 |

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{2}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{5}$$

In the formula, $N$ refers to the total number of data samples. Among them, formula (2) is the detection rate, which refers to the ratio of the normal data and the abnormal data of the correct classification to the total data. Formula (3) is the precision, which means that the number of attacks correctly divided into attacks is divided into the total proportion of attack data, which can reflect the ability of the model to identify the attack data. Equation (4) is the true positive rate, which represents the proportion of correctly identified attack data instances in all attack data. The higher the value of the above three evaluation indicators, the better the model effect. Formula (5) is a false positive rate, which refers to the ratio of normal data misclassification to the proportion of all attack data occupied by abnormal data. The lower the value, the better the model effect.

*4.3. Experimental Results Analysis.* Through the extraction and preprocessing fo the NSL-KDD algorithm set, which is then applied to the semisupervised learning model proposed in this paper, the performance of the proposed algorithm is compared with the spectral clustering algorithm, $K$-means algorithm and $K$- means + C4.5. As shown in Figures 5–7, the spectral clustering algorithm performs better than the $K$-means algorithm in terms of detection rate, accuracy, and true positive rate. The detection method of $K$-means + C4.5 is better than separate $K$-means or spectral clustering. Compared with other methods, the semisupervised learning model based on spectral clustering and random forest proposed in this paper is optimal in detection rate, precision, and true positive rate.

The false positive rate refers to the proportion of misclassification. The lower false positive rate is an important

Table 5: Confusion matrix.

| Prediction | | Actual | | |
|---|---|---|---|---|
| | | Positive | Negative | Total |
| Forecast result | Positive | TP | FP | Predicted as the amount of attack data |
| | Negative | FN | TN | Predicted as normal data |
| Total | | Actually attack number of data | Actually normal number of data | Total number of all data |

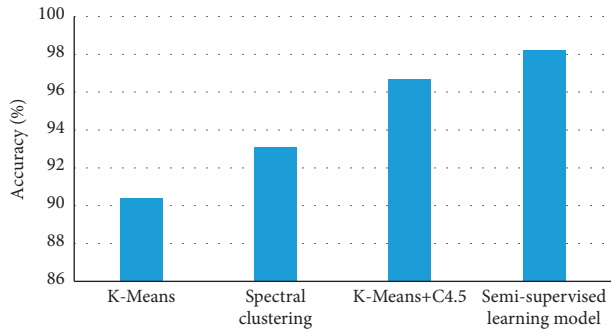


Figure 5: Comparison of accuracy of each algorithm.

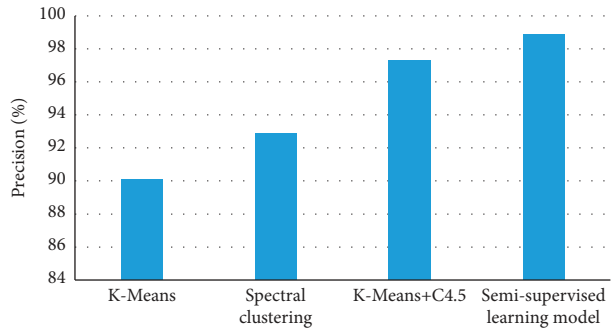


Figure 6: Comparison of precision of each algorithm.

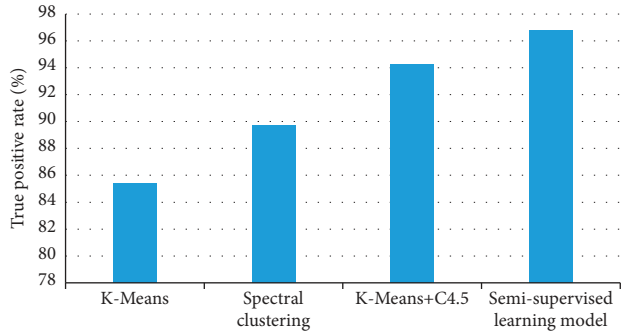


Figure 7: Comparison of true positive ratios of algorithms.

performance index for evaluating the detection algorithm. By comparing the above methods and calculating the average value, the experimental results are shown in Figure 8. The proposed semisupervised learning detection model has a lower false positive rate, which is basically consistent with the false positive rate of *K*-means + C4.5. The detection rate, accuracy, and true positive rate of the semisupervised learning model are higher than *K*-means + C4.5; therefore, the semisupervised learning detection model is more advantageous.

The experimental results show that the semisupervised learning model proposed in this paper has high accuracy,

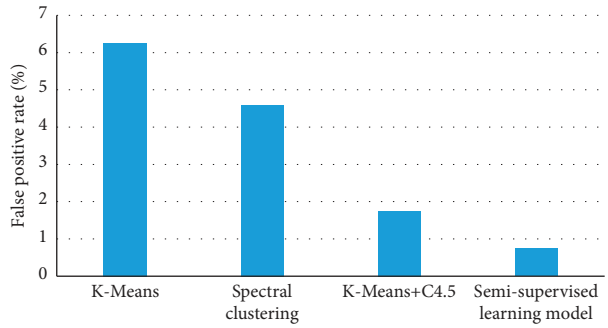


Figure 8: Comparison of false positive rate of each algorithm.

low false positive rate, and good performance. It is more suitable for detecting WEB DDoS attacks than other detection models.

According to the experimental results, the proposed method maintains a relative low false positive rate which is superior to unsupervised methods, and it can detect new types of attack network traffic data effectively. Additionally, the proposed method outperforms the hybrid method, K-means + C4.5, on all aspects of TPR, FPR, and precision.

## 5. Conclusion

In order to improve the detection rate of the existing WEB DDoS attack detection model, this paper proposes a semi-supervised learning model based on spectral clustering and random forest. First of all, due to the importance of flow characteristics to the detection scheme, we focus on it to select better features to be applied to the detection model proposed in this paper. Then, we analyze the spectral clustering algorithm and the random forest algorithm in detail. Based on the principle and its advantages, spectral clustering and random forest are combined to form a semisupervised learning WEB DDoS attack detection model. Finally, the experiment proposed in this paper is compared with other existing detection schemes to verify the paper. The proposed semisupervised learning model has a certain improvement in the detection rate while ensuring a low false positive rate and is more suitable for the detection of WEB DDoS attacks. In the future work, we will work on the improvement of the detection model and try some other machine learning methods in different manners.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This research was supported by the Guangdong Province Key Area R&D Program of China (Grant nos. 2019B010137004 and 2019B010136003), the National Natural Science Foundation of China (Grant nos. 61972108 and 62072131), the National Key Research and Development Plan (Grant no. 2018YFB0803504), Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019), the Science and Technology Projects in Guangzhou (Grant no. 202102010442), and the Science and Technology Project of Taizhou (2003gy15 and 20ny13).

## References

[1] D. Kaur and P. Kaur, "Empirical analysis of Web attacks," *Procedia Computer Science*, vol. 78, pp. 298–306, 2016.

[2] S. K. Ajagekar and V. Jadhav, "Study on web DDOS attacks detection using multinomial classifier," in *Proceedings of the IEEE International Conference on Computational Intelligence & Computing Research IEEE*, Chennai, India, December 2016.

[3] K. Ramasubramanian and A. Singh, "Machine learning theory and practices," *Machine Learning Using R*, Apress, Berkeley, CA, 2017.

[4] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee in *Proceedings of the 2008 ACM Conference on Emerging Network Experiment and Technology, CoNEXT 2008*, December 2008.

[5] R. A. Calix and S. Rajesh, "Feature ranking and support vector machines classification analysis of the NSL-KDD intrusion detection corpus," in *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference*, Palo Alto, California, May 2013.

[6] M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," *Procedia Engineering*, vol. 30, no. 4, pp. 1–9, 2012.

[7] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-means clustering and C4.5 decision tree algorithm," *Procedia Engineering*, vol. 30, pp. 174–182, 2012.

[8] Y. Chen, A. Abraham, and B. Yang, "Hybrid flexible neural-tree-based intrusion detection systems," *International Journal of Intelligent Systems*, vol. 22, no. 4, pp. 337–352, 2007.

[9] R. Cheng, R. Xu, X. Tang, V. S. Sheng, and C. Cai, "An abnormal network flow feature sequence prediction approach for ddos attacks detection in big data environment," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 95–119, 2018.

[10] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[11] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for internet of things via

ensemble classification," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5810–5818, 2021.

[12] Z. Tian, X. Gao, S. Su, and J. Qiu, "Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3901–3909, 2020.

[13] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-TTE: intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2659–2666, 2020.

[14] W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining svms with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127–140, 2014.

[15] J. E. Gaffney and J. W. Ulvila, "Evaluation of intrusion detectors: a decision theory approach," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pp. 50–61, Oakland, CA, USA, May 2001.

[16] X. Huang, Y. Ye, L. Xiong, S. Wang, and X. Yang, "Clustering time-stamped data using multiple nonnegative matrices factorization," *Knowledge-Based Systems*, vol. 114, pp. 88–98, 2016.

[17] M. N. Islam, M. Seera, and C. K. Loo, "A robust incremental clustering-based facial feature tracking," *Applied Soft Computing*, vol. 53, no. 53, pp. 34–44, 2017.

[18] P. G. Jeya, M. Ravichandran, and C. Ravichandran, "Efficient classifier for r2l and u2r attacks," *International Journal of Computer Application*, vol. 45, no. 21, pp. 28–32, 2012.

[19] J. Kang and S. Oh, "Anomaly intrusion detection based on clustering a data stream," *International Journal of Future Computer and Communication*, vol. 1, no. 1, pp. 17–20, 2012.

[20] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2017.

[21] M. Li, Y. Sun, Y. Jiang, and Z Tian, "Answering the min-cost quality-aware query on multi-sources in sensor-cloud systems," *Sensors*, vol. 18, no. 12, pp. 1–16, 2018.

[22] P.-C. Lin and J.-H. Lee, "Re-examining the performance bottleneck in a nids with detailed profiling," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 768–780, 2013.

[23] S.-W. Lin, K.-C. Ying, C.-Y. Lee, and Z.-J. Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," *Applied Soft Computing*, vol. 12, no. 10, pp. 3285–3290, 2012.

[24] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13–21, 2015.

[25] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–41, Article ID 12, 2015.

[26] W. Lee, S. J. Stolfo, and W. W. Mok, "A data mining framework for building intrusion detection models. "Security and Privacy"," in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pp. 23–30, Oakland, CA, USA, 1999.

[27] J. Qiu, Y. Chai, Y. Liu, Z. Gu, S. Li, and Z. Tian, "Automatic non-taxonomic relation extraction from big data in smart city," *IEEE Access*, vol. 6, pp. 74854–74864, 2018.

[28] A. Y. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Advances in Neural Information Processing Systems*, vol. 14, MIT Press, Cambridge, MA, USA, 2001.

[29] R. M. Saad, S. Manickam, and S. Ramadass, "Utilizing data mining approches in the detection of intrusion in ipv6 network: review & analysis," *International Journal on Network Security*, vol. 4, no. 1, pp. 35–39, 2013.

[30] Z. Tian, Y. Cui, L. An et al., "A real-time correlation of host-level events in cyber range service for smart campus," *IEEE Access*, vol. 6, pp. 35355–35364, 2018.

[31] Y. Wang, Z. Tian, H. Zhang, S. Shu, and W. Shi, "A privacy preserving scheme for nearest neighbor query," *Sensors*, vol. 18, no. 8, pp. 1–15, 2018.

[32] X. Wu, C. Zhang, R. Zhang, Y. Wang, and J. Cui, "A distributed intrusion detection model via non-destructive partitioning and balanced allocation for big data," *Computers, Materials & Continua*, vol. 56, no. 1, pp. 61–72, 2018.

[33] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," *Computers & Security*, vol. 94, Article ID 101863, 2020.

[34] K. Nasr, A. A.-E. Kalam, and C. Fraboul, "Performance analysis of wireless intrusion detection systems," in *Proceedings of the International Conference on Internet and Distributed Computing Systems*, pp. 238–252, Fujia, China, November 2012.

[35] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.

[36] Z. Tian, M. Li, M. Qiu, Y. Sun, and S. Su, "Block-DEF: a secure digital evidence framework using blockchain," *Information Sciences*, vol. 491, pp. 151–165, 2019.

[37] A. Alhussain, H. Kurdi, and L. Altoaimy, "A neural network-based trust management system for edge devices in peer-to-peer networks," *Computers, Materials & Continua*, vol. 59, no. 3, pp. 805–816, 2019.

[38] S. Li, Q. Zhang, X. Wu, W. Han, and Z. Tian, "Attribution classification method of APT malware in IoT using machine learning techniques," *Security and Communication Networks*, vol. 2021, pp. 1–12, Article ID 9396141, 2021.

[39] A. Amin, X.-H. Liu, I. Khan, P. Uthansaku, M. Forsat, and S. Sajad Mirjavadi, "A robust resource allocation scheme for device-to-device communications based on q-learning," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1487–1505, 2020.

[40] A. Badshah, A. Ghani, M. A. Qureshi, and S. Shamshirband, "Smart security framework for educational institutions using internet of things (iot)," *Computers, Materials & Continua*, vol. 61, no. 1, pp. 1–101, 2019.

[41] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, 2018.

[42] S. Li, Y. Li, W. Han, X. Du, M. Guizani, and Z. Tian, "Malicious mining code detection based on ensemble learning in cloud computing environment," *Simulation Modelling Practice and Theory*, vol. 113, Article ID 102391, 2021.

[43] S. Long, W. Long, Z. Li, K. Li, Y. Xia, and Z. Tang, "A game-based approach for cost-aware task assignment with QoS constraint in collaborative edge and cloud environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1629–1640, 2021.

[44] Z. Li, B. Chang, S. Wang, A. Liu, F. Zeng, and G. Luo, "Dynamic compressive wide-band spectrum sensing based on channel energy reconstruction in cognitive internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2598–2607, 2018.

[45] H. Yang, S. Li, X. Wu, H. Lu, and W. Han, "A novel solutions for malicious code detection and family clustering based on machine learning," *IEEE Access*, vol. 7, no. 1, pp. 148853–148860, 2019.