Khundrakpam Johnson Singh* and Tanmay De

Efficient Classification of DDoS Attacks Using an Ensemble Feature Selection Algorithm

https://doi.org/10.1515/jisys-2017-0472
Received September 14, 2017; previously published online December 1, 2017.

Abstract: In the current cyber world, one of the most severe cyber threats are distributed denial of service (DDoS) attacks, which make websites and other online resources unavailable to legitimate clients. It is different from other cyber threats that breach security parameters; however, DDoS is a short-term attack that brings down the server temporarily. Appropriate selection of features plays a crucial role for effective detection of DDoS attacks. Too many irrelevant features not only produce unrelated class categories but also increase computation overhead. In this article, we propose an ensemble feature selection algorithm to determine which attribute in the given training datasets is efficient in categorizing the classes. The result of the ensemble algorithm when compared to a threshold value will enable us to decide the features. The selected features are deployed as training inputs for various classifiers to select a classifier that yields maximum accuracy. We use a multilayer perceptron classifier as the final classifier, as it provides better accuracy when compared to other conventional classification models. The proposed method classifies the new datasets into either attack or normal classes with an efficiency of 98.3% and also reduces the overall computation time. We use the CAIDA 2007 dataset to evaluate the performance of the proposed method using MATLAB and Weka 3.6 simulators.

Keywords: Entropy, DDoS, classifier, MLP, information gain, chi-square, ensemble algorithm, feature selection.

1 Introduction

Denial of service (DoS) attacks [23] are carried out from a single compromised system having a malicious automated program to use up the resources (bandwidth, memory, CPU, etc.) of the target victim server. It attempts to interrupt the normal operations of an information system with the primary intention of harming the victim server. Legitimate clients trying to connect to the same server as the DoS attacks could not access the server resources. It is either due to exhaustion of the resources or denial by the server. A single compromised system consumes time to bring down a server, and can be detected easily through the accessing nature and could be prevented. In contrast, a distributed DoS (DDoS) attack [13] deploys a large number of compromised systems located in different geographical areas to perform the DoS attack to a particular target victim. These compromised systems are referred to as bots, and the network of bots is called botnets [13]. These attacks require less time to achieve their goal and make the server deny legitimate clients and even crash the server in the worst case.

There are mainly two types of DDoS attacks [2]: network layer DDoS attacks and application layer DDoS attacks. The network layer DDoS attack uses the open system interconnection (OSI) layer 3 or 4 protocols such as internet control message protocol (ICMP) to flood the victim server. The application layer DDoS attack deploys OSI layer 7 protocols such as hyper text transfer protocol (HTTP), domain name service (DNS), voice over internet protocol (VoIP), etc. to overwhelm the server, thereby exhausting the victim resources. This attack restricts legitimate clients from accessing the resources of the server. The DDoS attacks are detected by analyzing their anomalous traffic features, their accessing behavior, etc.

***Corresponding author: Khundrakpam Johnson Singh,** National Institute of Technology, 713209 Durgapur, West Bengal, India, e-mail: johnkh34@gmail.com
Tanmay De: National Institute of Technology, 713209 Durgapur, West Bengal, India

Selection of a single parameter to detect a DDoS attack is very limited to a particular type of DDoS attack and produce false positives. However, selection of too many parameters consumes more resources in terms of network and time of computation. In addition, selection of a weak feature extraction algorithm will omit the most useful parameters.

Concerning the above-stated problems, we proposed a new approach to detect DDoS attacks. The main contribution of our work can be summarized as follows:

- Using CAIDA 2007 datasets [3], we build the attack and the normal client profile. We determine 16 features that influence the detection of DDoS attacks. These features include the duration, protocol type, source port, destination port, frame number, frame length, capture length, payload length, hop limit, urgent, inter-arrival time, probability of uniqueness of the IP address, acknowledgment flag, synchronous flags, time-to-live (TTL) field, and sequence number. Selection of specific parameters from the above 16 features will affect the efficiency in detecting DDoS attacks.
- We propose an ensemble feature selection technique [12] to select the most effective features from the given 16 features.
- We introduce the application of various classifiers, such as mutilayer perceptron (MLP) with the back-propagation method [25], naive Bayes [11], random forest [1], and radial basis function (RBF) network [5], to classify the dataset into attack and normal classes.
- We finally select the classifier with maximum accuracy for testing unknown attack datasets with the finally selected features.

2 Related Works

Yatagai et al. [26] proposed two algorithms to detect an HTTP-GET flood attack based on the browsing order of pages and correlation with browsing time to the page information size. They considered the idea that an attack from compromised clients induced by the same virus or bot observes the same browsing order of pages continually at the server. Their article presented the concept that an attacker browses a web page for a shorter duration than regular users. However, research on the work of the bot and attacking machine shows that their code can be modified to access the pages randomly and browse the web page for a longer time.

Ko et al. [9] proposed an anti-DDoS mechanism based on flow-based network forwarding technique. They analyzed the first packet of a flow and sent the rest of the packets based on the behavior of the first packet. The authors used the idea that bot or attackers could not solve the completely automated public turing test to tell computer and human apart (CAPTCHA) problem, and hence could easily differentiate legitimate clients from attackers. However, sending of packets based on the nature of the flow of the first packet cannot be trusted, as attackers can, at any time, manipulate the intermediate nodes. The use of the CAPTCHA technique consumes bandwidth, is time consuming, and annoys legitimate clients accessing the web pages.

Tsai et al. [22] proposed an early warning system for DDoS attack detection based on the rationale of the time delay neural network. A multilayer architecture was designed by monitoring each node in the topology by the deployment of detectors. The authors considered the concept that distinct signatures are embedded in the traffic that is generated by the freely available attackers. Thus, this particular signature could be detected easily by intrusion detection system (IDS). In the initial phase of the attack, there is excellent communication between the attacker and the thousands of zombie computers. However, installing detectors at each node will increase the network consumption and cost. The attack traffic rarely consists of a particular signature in their traffic, and hence contrasts the concept introduced in the article. An attacker will never communicate with the zombie computers directly; it uses compromised machines to recruit the zombie computers on its behalf.

Thapngam et al. [20] proposed a model of DDoS attack detection based on the behavior of the incoming traffic. The authors considered the predictable rate attacks that include analyzing the features that have a

constant rate, periodical rate, or monotonically increasing rate in the traffic flow. They used the Pearson correlation coefficient theorem with the predictable rate attack features to detect a normal class, attack class, and flash crowds. However, the article did not concern with non-predictable rate attacks where the features have random behaviors. In general, the DDoS attack rarely has a constant rate of features, and detecting the attacks by being concerned with only one feature will not be efficient.

Kalkan and Alagöz [7] proposed a mechanism known as the ScoreForCore for detecting and filtering DDoS attack packets. In this method, a score is calculated for every incoming packet using the nominal profile and the current profile of the traffic. The method identifies whether the traffic is an attack or is normal based on the calculated score of the incoming packets. The nominal profile is created by counting the number of packets during a normal period, and the attack profile is created during the attack period. The method considers attributes such as IP address, port number, protocol type, packet size, TTL value, and transmission control protocol (TCP) flag for the detection.

Xiao et al. [24] proposed a correlation-based detection of DDoS attacks against a data center. It makes use of the correlation information of flow in the data center. To reduce the overhead caused by the size of the training dataset, a mechanism based on K-nearest neighbor with correlation (CKNN) and r-polling model is utilized. The CKNN model is more suitable than the KNN model in classifying network traffic even with a high noise signal. The method is tested against the KDD'99 dataset.

3 Feature Selection

There are many features from the traffic flow that changes their characteristics with time, and these features could be considered for the analysis of DDoS attacks. In this article, we list some of them, which are found through the analysis of various attacks and normal traffic. These features include duration, protocol type, source port, destination port, frame number, frame length, capture length, payload length, hop limit, urgent, inter-arrival time, probability of uniqueness of the IP address, acknowledgment flag, synchronous flags, TTL field, and sequence number. Selection of all these 16 features will not only consume time but also system resources in terms of CPU or memory. Thus, selection of specific and more appropriate features for classification of traffic is important in order to provide better efficiency with minimum error.

3.1 Feature Selection Algorithm and Classification

In this article, we will not use all the 16 features stated above. We will find the most efficient features that can differentiate the two different classes (attack and normal). For the purpose of feature selection, we will use the ensemble feature algorithm.

3.1.1 Information Gain Algorithm

This algorithm uses the concept of information theory [17]: the higher the entropy, the more the content of the information. The algorithm determines which features in a given set of training dataset elements are more appropriate in differentiating between the classes to be learned. In this article, we use two main classes: an attack class and a normal class. Each feature is considered as the parent node of the decision tree, as shown in Figure 1. The parent node consists of the α number of attack classes and the β number of normal classes. The child node represents the classes corresponding to the particular value of the attribute considered. In general, the entropy is defined by Eq. (1). To calculate the information gain of a particular attribute, we consider the entropy of the parent node as given by Eq. (2); the entropy of the child nodes is given by Eq. (3). After finding the entropy of the parent and child nodes, we calculate the information gain

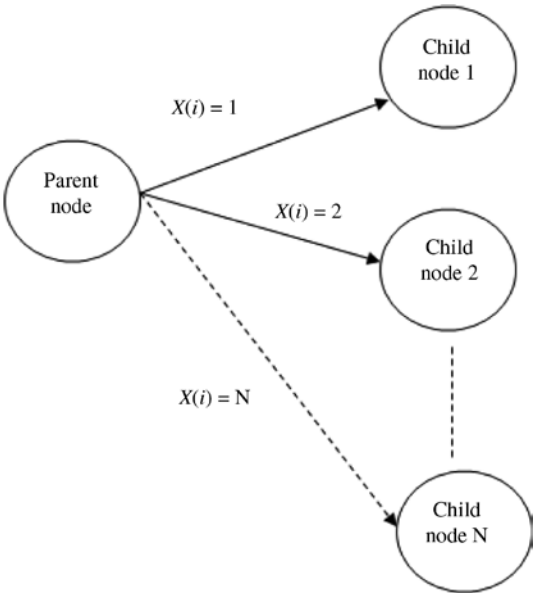

Figure 1: Relation of Parent Node and Child Nodes in a Decision Tree with Respect to the Value of the Feature Provided by $X(i)$.

Table 1: Definition of Notations.

Notations	Definition
P_i	Probability of class i
N	Number of attribute values considered within the optimized value ranges
α	Number of attack class in the parent node
β	Number of normal class in the parent node
α_c	Number of attack class in the child node corresponding to the value of the attribute
β_c	Number of normal class in the child node corresponding to the value of the attribute
$E(P)$	Entropy of the parent node
$E(C)$	Entropy of child node

of the respective features given by Eq. (4). The definition of the notation used in the article is defined in Table 1.

$$\text{Entropy} = \sum_{i=1}^{N} -P_i \log_2 P_i. \tag{1}$$

$$E(P) = -\left[\left(\frac{\alpha}{\alpha+\beta}\log_2\frac{\alpha}{\alpha+\beta}\right)+\left(\frac{\beta}{\alpha+\beta}\log_2\frac{\beta}{\alpha+\beta}\right)\right]. \tag{2}$$

$$E(C) = -\sum_{i=1}^{N}\left[\frac{\alpha_c}{\alpha_c+\beta_c}\log_2\frac{\alpha_c}{\alpha_c+\beta_c}+\frac{\beta_c}{\alpha_c+\beta_c}\log_2\frac{\beta_c}{\alpha_c+\beta_c}\right]. \tag{3}$$

$$IG_{\text{Attribute}} = E(P) - \frac{1}{N}E(C). \tag{4}$$

We now consider the 16 features of the clients that are destined to the particular victim machine IP address. The computation of the information gain given by Eq. (4) is illustrated in Table 2.

3.1.2 χ^2 Algorithm

The chi-square (χ^2) measurement [21] is utilized to test the freedom of two features by determining a score to measure the degree of independence of these two features.

$$\chi^2 = \sum \frac{(F_{\text{observed}} - F_{\text{expected}})^2}{F_{\text{expected}}}. \tag{5}$$

It is also observed that $\sum F_{\text{observed}} = \sum F_{\text{expected}} = \text{total frequency}$, where F_{observed} is the frequency of either attack or normal for each child and F_{expected} is calculated by multiplying both the total rows and columns, and then dividing by the total frequency. The result obtained from the calculation will show by how much each feature is related in predicting the status (either attack or normal) of the clients. A large value of the χ^2 demonstrates a high dependency in the relationship. The calculated value of χ^2 for all the 16 features is given in Table 2.

3.1.3 Gain Ratio

The gain ratio [4] is the modification to enhance the bias of information gain toward features with a significant diversity value. Gain ratio conquers the issue with information gain by considering the number of branches that would come about before doing the split. It revises information gain by considering the intrinsic characteristic data of a split.

$$\text{Gain Ratio}\ (y, x) = \frac{IG_{\text{Attributes}}(y, x)}{\text{Intrinsic Value}\ (x)}, \tag{6}$$

where

$$\text{Intrinsic Value}(x) = -\sum \frac{\text{frequency of child}}{\text{frequency of parent}} \times \log_2 \frac{\text{frequency of child}}{\text{frequency of parent}} \tag{7}$$

The computed value of gain ratio using Eqs. (6) and (7) is given in Table 2.

Table 2: Ranking of 16 Features with Various Feature Selection Algorithms.

Attribute	information Gain	χ^2	Gain ratio	SVM	Correlation ranking filter	ReliefF	Symmetrical uncertainty ranking filter
Duration	0.726	0.699	0.674	0.742	0.643	0.762	0.637
Protocol_type	0.163	0.112	0.199	0.149	0.153	0.233	0.119
src_port	0.246	0.279	0.218	0.091	0.251	0.227	0.238
dst_port	0.272	0.235	0.273	0.184	0.243	0.241	0.278
frame_no	0.117	0.118	0.134	0.083	0.115	0.137	0.206
frame_len	0.232	0.217	0.364	0.117	0.197	0.271	0.174
capture_len	0.034	0.042	0.117	0.073	0.063	0.073	0.184
payload_len	0.274	0.207	0.174	0.179	0.137	0.054	0.341
hop_limit	0.102	0.143	0.147	0.138	0.142	0.548	0.121
urgent	0.237	0.187	0.211	0.164	0.078	0.134	0.224
M_T	0.682	0.763	0.731	0.883	0.864	0.743	0.669
Prob_{IP}	0.632	0.703	0.773	0.786	0.832	0.832	0.706
A_{flag}	0.487	0.521	0.435	0.598	0.504	0.558	0.619
S_{flag}	0.513	0.544	0.576	0.607	0.612	0.662	0.574
TTL	0.9	0.917	0.887	0.872	0.943	0.753	0.665
Sq_{flag}	0.542	0.622	0.604	0.632	0.664	0.582	0.439

3.1.4 Other Feature Selection Methods

The other feature selection methods used in the article are ReliefF [15], support vector machine (SVM) [14], correlation ranking filter [18], and symmetrical uncertainty ranking filter [8]. ReliefF calculates the ranks and weights of features for the given data input using the ReliefF algorithm. It is a feature selection strategy that utilizes ceaseless testing to assess the value of a component to recognize between the closest hit and closest miss. We determine a user-defined threshold, and that weight of features that surpasses this threshold value is chosen as the essential feature. It works on both discrete and continuous data class. SVM evaluates the worth of the features by using the SVM classifier [16]. Features are ranked by the square of the weights assigned by the SVM classifier. The correlation ranking filter evaluates the worth of a feature by calculating Pearson's coefficient [27]. The symmetrical uncertainty ranking filter evaluates the worth of a feature by measuring the symmetrical uncertainty with respect to the class. The calculation of the SVM, correlation ranking filter, ReliefF, and symmetrical uncertainty ranking filter for the 16 features from the CAIDA 2007 dataset is given in Table 2.

3.1.5 Ensemble Methods

Instead of selecting the features based on the ranking of a single feature selection algorithm, we create an ensemble approach, as shown in Figure 2. This method consists of the combined workings of the seven feature selection algorithms. We calculated the average of each algorithm as shown in Table 3 and computed the threshold value, h, which is obtained by taking the average of the seven feature selection algorithms. In Figure 2, the value of the feature (F value), which is smaller than the threshold value, is dropped; otherwise, the feature is considered.

3.2 Description of the Selected Features

3.2.1 Inter-arrival Time

Inter-arrival time represents the difference in time between any two consecutive requests from same IP address or between two requests from different IP addresses. The server measures the inter-arrival time for various users at random time intervals. The inter-arrival time between any two requests is calculated in seconds by

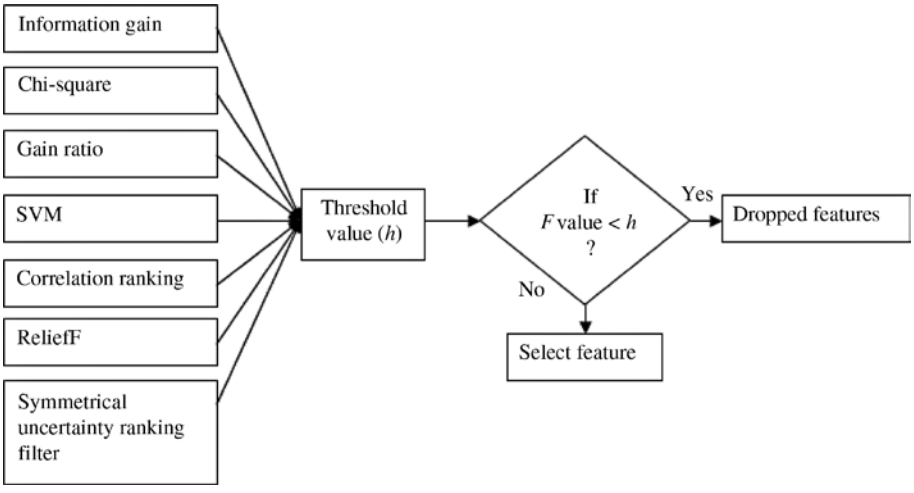

Figure 2: Selection of Features Through Ensemble Feature Selection Methods.

Table 3: Calculation of Threshold Value from Various Feature Selection Methods.

Feature selection methods	Average value	Threshold value, *h*
Information gain	0.375294	0.390235
χ^2	0.390941	
Gain ratio	0.402588	
SVM	0.386588	
Correlation ranking filter	0.386412	
ReliefF	0.406175	
Symmetrical uncertainty ranking filter	0.383647	

subtracting the arrival time of the later request from that of the former one. The inter-arrival time in case of a normal and attack scenario is illustrated in Figure 3A and B, respectively.

3.2.2 Probability of Uniqueness of the IP Address

The probability of uniqueness of the IP address represents the number of occurrence of an IP address during an interval. It is calculated by dividing the number of occurrence of a particular IP address during a time interval by the total number of IP addresses that occur in that range. Figure 4A and B represent the probability of uniqueness of the IP address in normal and attack scenarios.

3.2.3 Acknowledgment and Synchronous Flags

The acknowledgment flag refers to the indication of a successful receipt of the packets sent. The flag is set to 1 for the successful receipt of packets and 0 for an unsuccessful receipt of packets. The synchronous flag refers to the indication that a connection is initiated. The flag value is set to 1 for initiation of a connection between the client and server. The flag value is set to 0 when no connection is initiated.

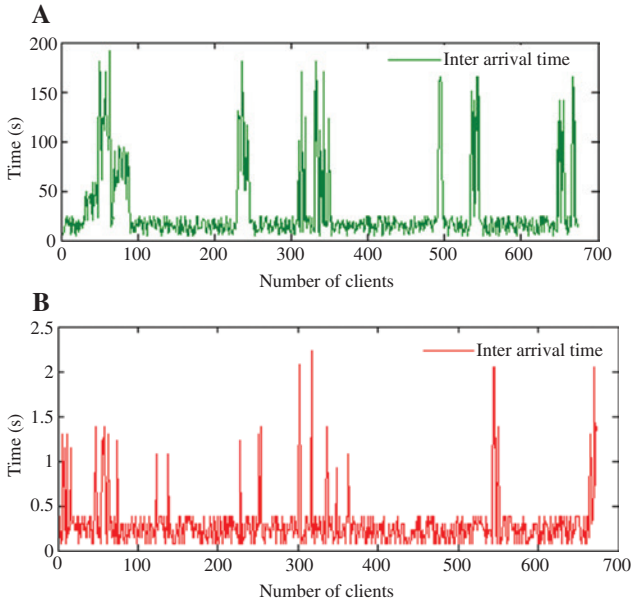

Figure 3: Inter-arrival Time in Case of (A) a Normal and (B) an Attack Scenario.

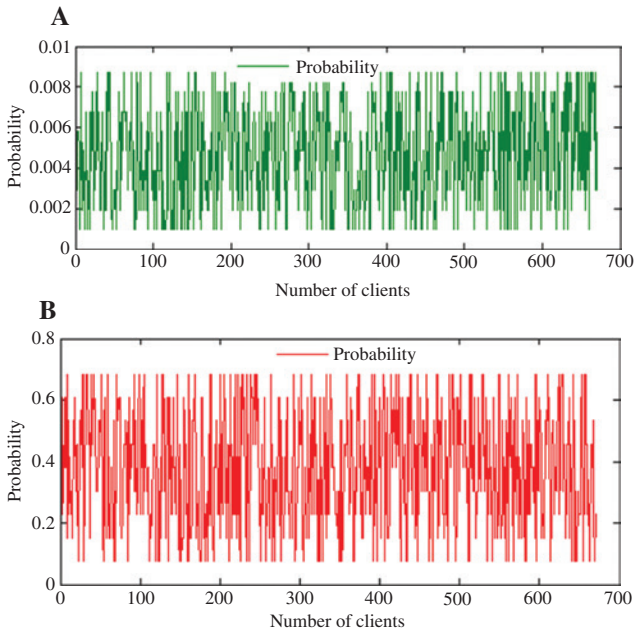

Figure 4: Probability of Uniqueness of the IP Address in (A) a Normal and (B) an Attack Scenario.

3.2.4 TTL, Sequence Number, and Duration

TTL refers to the maximum number of routers a packet from a source can traverse before either reaching the destination or being dropped. Figure 5A and B refer to the variations in the values of the TTL field in case of an attack and normal scenario, respectively. Sequence number relates to the counter that tracks how much data are sent and the sequence of the arrival of the packets. Duration refers to the total time taken by a specific client within a stipulated time frame to carry out the attack. It is given by the difference of time between the first instance of the client sending a request to a particular server and the time of sending a request from the same client at the end of the time frame.

3.3 Analysis of Features

The primary objective of a DDoS attack is to make the victim server refuse requests from legitimate clients. Here, the number of compromised clients is an important criterion to be considered. We divide the number of clients in the attack in two cases, as given below:

Case 1: When the number of DDoS attacking clients is large, the attackers can send the request as that in a normal scenario. However, these could be easily detected by setting the limitation in the number of

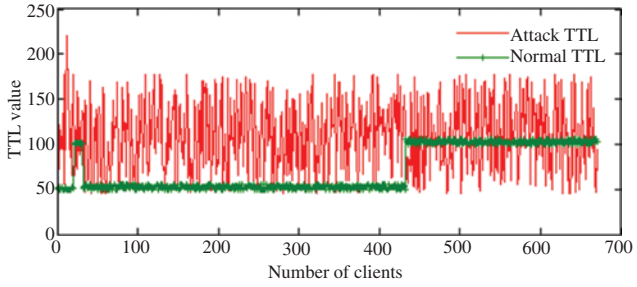

Figure 5: Variations in the Values of the TTL Field in Case of an Attack and a Normal Scenario.

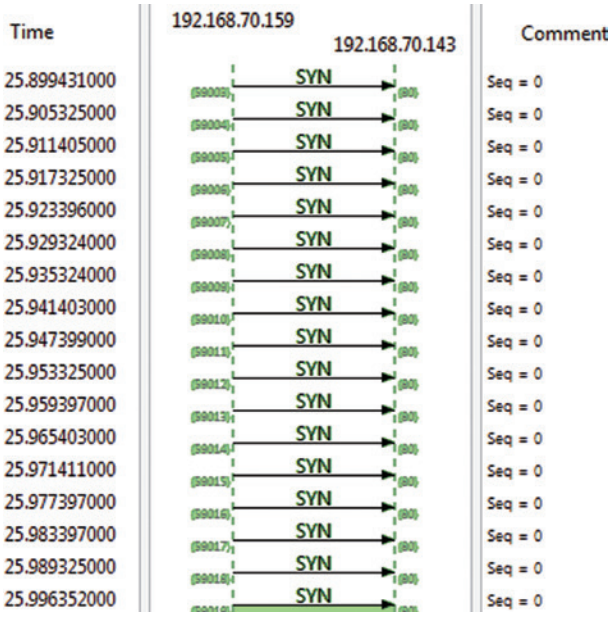

Figure 6: Nature of an Attacking Client When Accessing a Server.

concurrent connections. The Apache server can add the constraints to the number of clients connecting to the server simultaneously.

Case 2: When the number of DDoS attacking clients is medium or low, we are considering a mild attack. When the number of concurrent connections is below the maximum limit (threshold) raised by the server, it bypasses the connections. As this condition is not an offense against the threshold value, the attackers can bypass the first filtering rule. In this article, we consider the meek DDoS attack. The parameters discussed in the article are taken for a mild attack.

We consider an attack size of 20,000 clients that are accessing the victim target server. In this case, the inter-arrival time of any two consecutive request packets must be minuscule. The probability of the presence of an IP address within the given time window must be large. In the case of an attack, the acknowledgment flag is seldom set; the synchronous flag is configured to start the connection. The value of the TTL field in an attack packet is larger than normal, and finally, the sequence numbers of the attacking packets are not set. Figure 6 illustrates the nature of the flow of traffic in a Slowloris [13] attack from IP address 192.168.70.159 accessing the Apache server with IP address 192.168.70.143.

3.4 Selection of the Classifying Model

In this article, we deploy MLP, naive Bayes, RBF network, and random forest, which are machine learning classifiers for training the common features along with the target. It maps a set of input data to the set of output data. To select the most effective classifier, we plot the receiver operator characteristic (ROC) curve [19] using Weka 3.6 [6]. Figure 7 provides the comparison of ROC curves for various classifiers. It is observed that MLP has a more efficient ROC curve. In this article, we select the MLP classifier for the final classification. We further integrate previous ensemble feature selection algorithms with the MLP classification.

4 Results and Discussion

We simulate the throughput of the incoming traffic in case of an attack and normal scenario with the NetSim simulator. In the simulation, Link_1 is the victim target server and Link_3 is the server where no attack occurs.

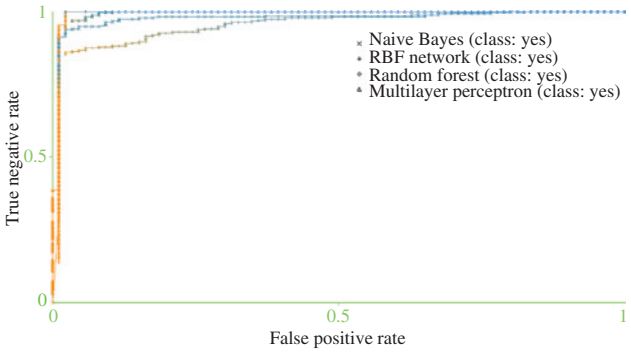

Figure 7: Comparison of ROC Curve for Various Classifiers.

The throughput generated in Link_1 and Link_3 during a mild attack and a normal day are shown in Figure 8A and B, respectively. These throughput factors could not, on their own, distinguish between an attack and normal mode, as, during a busy day, the throughput of a traditional server can be increased.

We will now input the 16 features considered in the article to the MLP classifier and obtain the confusion matrix result, as shown in Figure 9A, as generated by the Matlab simulator. Similarly, we consider the six selected features and input them to MLP and construct the confusion matrix result as shown in Figure 9B.

The two models are compared in terms of accuracy, specificity, sensitivity, time consumed, and root mean square error (RMSE), as shown in Table 4. To calculate the accuracy, specificity, and sensitivity from the confusion matrix in Figure 9A and B, we use Eqs. (8), (9), and (10), respectively:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d}, \tag{8}$$

$$\text{Specificity} = \frac{a}{a+b}, \tag{9}$$

$$\text{Sensitivity} = \frac{d}{c+d}, \tag{10}$$

where a, b, c, and d are given in Table 5.

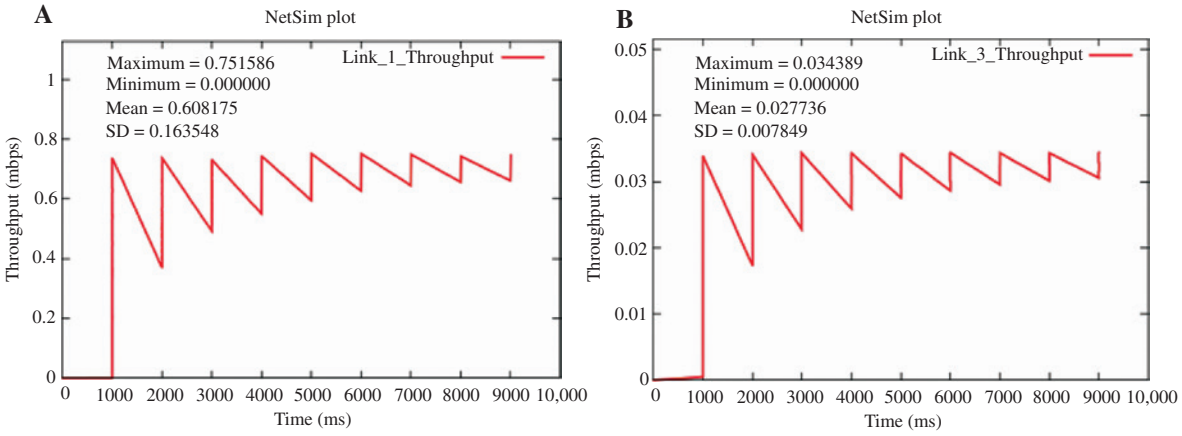

Figure 8: Throughput for (A) a DDoS Attack and (B) a Single Normal Connection.

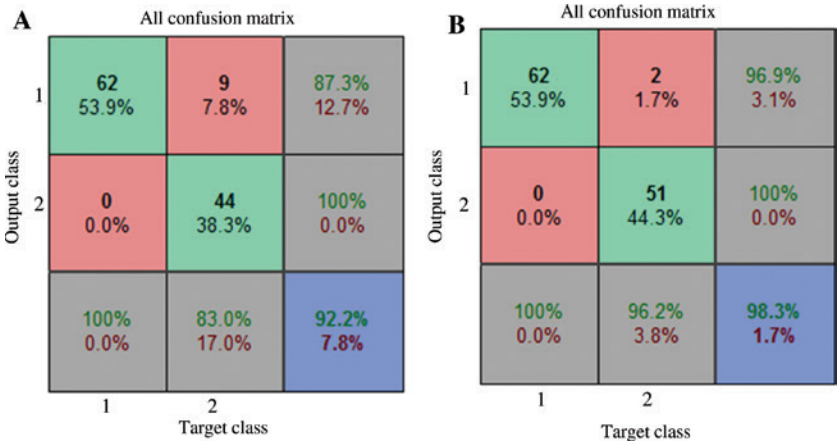

Figure 9: Confusion Matrix Result (A) Before and (B) After the Application of the Information Gain Algorithm.

Table 4: Comparison of MLP Model Before and After the Proposed Feature Selection Method.

Model	Accuracy	Specificity	Sensitivity	Time Consumed	RMSE
MLP (before)	92.2%	87.3%	100%	0.24 s	0.1222
MLP (after)	98.3%	96.9%	100%	0.09 s	0.089

Table 5: General Confusion Matrix Result.

	Actual	
Predicted	Negative	Positive
Negative	*a*	*b*
Positive	*c*	*d*

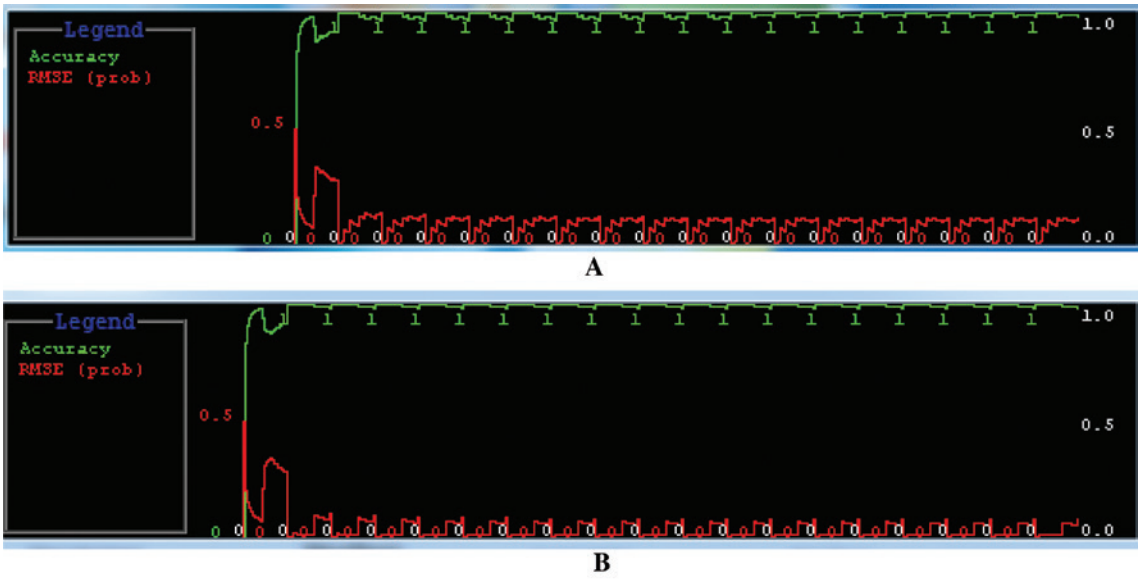

Figure 10: Accuracy and RMSE Comparison (A) Before and (B) After Filtering the Features.

Table 6: Comparison of the Proposed Method with Existing Techniques.

Methods	Accuracy	Specificity	Sensitivity
Proposed method	98.3%	96.9%	100%
Yatagai et al. [26]	88%	N/A	N/A
Tsai et al. [22]	76.5%	84%	73%
Thapngam et al. [20]	94%	N/A	N/A
Kalkan and Alagöz [7]	96%	98%	93%
Xiao et al. [24]	92.3%	N/A	N/A

N/A, not available.

A comparison between the two models indicates that the use of MLP after the ensemble feature selection algorithm provides more accuracy and specificity. The design provides lower computation overhead and has a smaller value of RMSE than the other model. We also run Weka 3.6 with the incremental naive Bayes classifier [10] to compare the accuracy of the two models with their respective probability of RMSE generated as shown in Figure 10A and B.

From Figure 10A and B, it is observed that the accuracy graph after the application of the ensemble feature selection algorithm shifts more toward the perfect curve, i.e. toward the value 1.0. Similarly, the RMSE value after the ensemble algorithm shifts down toward the minimum error value, i.e. 0. This comparison shows that application of the ensemble feature selection algorithm plays a significant role not only in increasing the accuracy of DDoS attack detection but also in reducing the error rate.

Table 6 provides a comparison of the accuracy of detection of the proposed DDoS attack mechanism with some of the existing techniques.

5 Conclusions

In this article, an approach to detect DDoS attacks based on a feature selection scheme is proposed. The use of an ensemble feature selection algorithm helps in the efficient selection of useful features. The threshold value obtained from the ensemble algorithm decides whether a feature is to be selected or dropped. The proposed method provides a better accuracy of 98.3% in classification than in the use of all features. In this article, we analyzed four classification algorithms and selected MLP as it provides better accuracy than the other three. The proposed model produces less computation overhead in terms of time with a lower RMSE value. The proposed attack detection mechanism is applicable in most of the available DDoS attack datasets, and hence could be applied to real-time incoming traffic. In the future, the proposed method can be extended by integrating the DDoS attack detection scheme with a blacklisting and prevention plan using IP table rules.

Bibliography

[1] W. T. Aung, Y. Myanma and K. H. M. S. Hla, Random forest classifier for multi-category classification of web pages, in: *Proceeding of the IEEE Asia-Pacific Conference on Services Computing*, Singapore, Singapore, 7–11 December, 2009.

[2] H. Beitollahi and G. Deconinck, Tackling application-layer DDoS attacks, *Proc. Comput. Sci.* **10** (2012), 432–441.

[3] Center for Applied Internet Data Analysis, *The CAIDA UCSD "DDoS Attack 2007" Dataset*, Available at: http://www.caida.org/data/passive/ddos-20070804_dataset.xml. Accessed 16 January, 2015.

[4] M. A. Hall and L. A. Smith, Practical feature subset selection for machine learning, in: *Proceedings of the 21st Australian Computer Science Conference*, Berlin, Germany, pp. 181–191, 1998.

[5] T. Ince, S. Kiranyaz and M. Gabbouj, Evolutionary RBF classifier for polarimetric SAR images, *Expert Syst. Appl.* **39** (2012), 4710–4717.

[6] K. Jaswal, P. Kumar and S. Rawat, Design and development of a prototype application for intrusion detection using data mining, in: *Proceeding of the 4th International Conference on Infocom Technologies and Optimization*, Noida, India, 2–4 September, 2015.

[7] K. Kalkan and F. Alagöz, A distributed filtering mechanism against DDoS attacks: ScoreForCore, *Comput. Netw.* **108** (2016), 199–209.
[8] S. S. Kannan and N. Ramaraj, A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm, *Knowl. Based Syst.* **23** (2010), 580–585.
[9] N.-S. Ko, S.-K. Noh, J.-D. Park, S.-S. Lee and H.-S. Park, An efficient anti-DDoS mechanism using flow-based forwarding technology, in: *9th International Conference on Optical Internet (COIN), 2010*, pp. 1–3, 11–14 July, 2010.
[10] S. Kotsiantis, Increasing the accuracy of incremental naive Bayes classifier using instance based learning, *Int. J. Control Autom. Syst.* **11** (2013), 159–166.
[11] S. Kotsiantis, Integrating global and local application of naive Bayes classifier, *Int. Arab J. Inform. Technol.* **11** (2014), 300–307.
[12] S. M. Lee, D. S. Kim, J. H. Lee and J. S. Park, Detection of DDoS attacks using optimized traffic matrix, *Comput. Math. Appl.* **63** (2012), 501–510.
[13] S. McGregory, Preparing for the next DDoS attack, *Netw. Secur.* **2013** (2013), 5–6.
[14] S. Paul, M. Magdon-Ismail and P. Drineas, Feature selection for linear SVM with provable guarantees, *Pattern Recognit.* **60** (2016), 205–214.
[15] O. Reyes, C. Morell and S. Ventura, Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context, *Neurocomputing* **161** (2015), 168–182.
[16] F. Rubio, J. Martínez-Gómez, M. J. Flores and J. M. Puerta, Comparison between Bayesian network classifiers and SVMs for semantic localization, *Expert Syst. Appl.* **64** (2016), 434–443.
[17] A. Sadri, Y. Ren and F. D. Salim, Information gain-based metric for recognizing transitions in human activities, *Pervas. Mobile Comput.* **38** (2017), 92–109.
[18] N. Sánchez-Maroño, A. Alonso-Betanzos A and M. Tombilla-Sanromán, Filter methods for feature selection – a comparative study, in: H. Yin, P. Tino, E. Corchado, W. Byrne and X. Yao (Eds.), *Intelligent Data Engineering and Automated Learning – IDEAL 2007*, Springer, Berlin, Heidelberg, 2007.
[19] C. M. Schubert, M. E. Oxley and K. W. Bauer, A comparison of ROC curves for label-fused within and across classifier systems, in: *Proceeding of the 7th International Conference on Information Fusion*, Philadelphia, PA, USA, 25–28 July, 2005.
[20] T. Thapngam, Y. Shui, W. Zhou and G. Beliakov, Discriminating DDoS attack traffic from flash crowd through packet arrival patterns, in: *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Shanghai, China, pp. 952–957, 10–15 April, 2011.
[21] I. S. Thaseen and C. A. Kumar, Intrusion detection model using fusion of chi-square feature selection and multi class SVM, *J. King Saud Univ. Comput. Inform. Sci.* **29** (2017), 462–472.
[22] C.-L. Tsai, A. Y. Chang and M.-S. Huang, Early warning system for DDoS attacking based on multilayer deployment of time delay neural network, in: *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Darmstadt, Germany, pp. 704–707, 15–17 October, 2010.
[23] B. Wang, Y. Zheng, W. Lou and Y. T. Hou, DDoS attack protection in the era of cloud computing and software-defined networking, *Comput. Netw.* **81** (2015), 308–319.
[24] P. Xiao, W. Qu, H. Qi and Z. Li, Detecting DDoS attacks against data center with correlation analysis, *Comput. Commun.* **67** (2015), 66–74.
[25] J. Yang, X. Zeng and S. Zhong, Computation of multilayer perceptron sensitivity to input perturbation, *Neurocomputing* **99** (2013), 390–398.
[26] T. Yatagai, T. Isohara and I. Sasase, Detection of HTTP-GET flood attack based on analysis of page access behavior, in: *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2007, PacRim 2007*, Victoria, BC, Canada, pp. 232–235, 22–24 August, 2007.
[27] H. Zhou, Z. Deng, Y. Xia and M. Fu, A new sampling method in particle filter based on Pearson correlation coefficient, *Neurocomputing* **216** (2016), 208–215.