An Ensemble Framework for Flow-Based Application Layer DDoS Attack Detection Using Data Mining Techniques

K. Munivara Prasad, V. Samba Siva, J. Nagamuneiah and Siddaiah Nelaballi

Abstract The large number of requests flow exceeds the capacity of the target server drives to denial in the service to the legitimate users. Due to the server's oversized prospective, the flooding requests increase the server capacity generated by the malicious attackers from distributed environment defining the distributed denial of service attack. From the contemporary literature it is evident that applying the knowledge gained from the findings of previous request distributions is a suitable strategy to block the DDoS attacks. This strategy's key limitation is frisking to detect the new patterns of request flooding excavated by the attacker at the server from the previous knowledge on earlier attack distributions patterns. Therefore, this paper explains a novel trained ensemble classifier with new features which reflects in the traffic flow properties, so that, the traffic flow shows distribution diversity from each other which is considered and attached to individual classifiers. Ensemble classifier and AdaBoost are used to detect the flow by discovering the distribution resemblance involved in the multiple classifiers in the ensemble classification model. The experiment worked out on the voluminous traffic flow with visible distribution variety.

Keywords DDoS attack · Ensembles approach · K-S test and application layer DDoS attacks

1 Introduction

Nowadays the Internet plays a major role in human life in various activities and allows to do all the day to day activities online which attracts the attackers to compromise the network and user services. The Denial of Service (DoS) attack [1] is a malicious attempt by a single person to compromise the network resources which are not being accessed by the authorized person. If it is done by a group of people it is called as Distributed DoS attack (DDoS). One of the main threats to the internet applications

K. M. Prasad (✉) · V. S. Siva · J. Nagamuneiah
Chadalawada Ramanamma Engineering College, Tirupati, India
e-mail: prasadkmv27@gmail.com

S. Nelaballi
S R E S Group of Institutions, Tirupati, India

S. Fong et al. (eds.), *ICT Analysis and Applications*, Lecture Notes in Networks and Systems 93, https://doi.org/10.1007/978-981-15-0630-7_2

is the Application layer DDoS [2] attacks where all the user applications and services are targeted.

From the literature it is evident that several approaches were developed to detect the DDoS attacks, but each had its own drawbacks and advantages, but these methods failed to maintain consistent results when the traffic is from diversified traffic. Ensemble-based classifiers are used in this paper to maintain consistent results even though the traffic is from the diversified network.

2 Related Work

Though the detection method and defense measure have been widely researched the complexity of the DDoS attack is higher and the size of the DDoS attack is much larger than before. Paper [3] introduced several public datasets used in the recent years. Different types of DDoS attack datasets were presented in the paper.

The similarities of all the datasets were the large number of attributes and information, which posed a great challenge to detect the attacks among massive information. For better performance to process the huge amount of information data mining method has been researched to detect the DDoS attack.

In paper [4], two kinds of data mining methods, MLP and Rand forest method were applied to detect the DDoS attack. Both the methods were proven to detect the DDoS attacks while the consuming time and computing cost were high after experiment verification because of the high amount of dataset and lots of attributes used in this experiment.

To detect the DDoS attack with a huge amount of data, methods on reducing the amount of data and advanced method to improve the accuracy need to be researched. Different ranking methods, Info gain, gain ratio, and chi-squared were implemented in paper [4] in order to get more important attributes. The time taken in build model was saved and the detecting rate was improved after the one third selection of the voted ranking while the one third ranking whether can contain the whole Information need to be considered. And further improvement also needed to be done.

In paper [5], three different data mining methods Bagging, Rand forest, and k-NN were applied. The final result was voted among the three heterogeneous methods. Though the accuracy was improved according to the paper; the TNR was not the best compared with others. Normally, voting among different methods always leads to the middle value rather than the best which may lead to the detecting rate not being stable.

ARM was applied to select the important features in paper [6], and two datasets were experimented in this paper. It showed that accuracy to detect the attack was improved but the accuracy to identify the normal events was deceased. It makes sense in identifying the attack to some extent but still needed to improve the whole ability to identify both the normal and attack events.

The large amount of data needs to be processed in DDoS attack detection, but little error rate even means many attacks were incorrectly detected. Though some of them

have contributed in improving the detection rate of DDoS attack to some extent, few paper majors in both improving the detection rate and reducing the amount of data at the same time. This paper aimed at improving the accuracy of DDoS detection by using ensemble data mining technology. The main target of this work is to reduce the unrelated data and improve the accuracy in detecting DDoS attacks at the same time.

3 Proposed Work

The proposed method includes the attack detection at the flow level rather than the request level. The dataset consists of attack and normal which is considered as the input for the process and each corpus is processed separately. The collection of normal requests from the input corpus is grouped as sessions with fixed time. The input dataset is now converted as session dataset. The sessions are grouped as the clusters using k-means cluster algorithm based on the session begin times. The clusters are grouped as the absolute time interval (ati) and the absolute time interval is defined as session begin intervals, session completion intervals, page access begin intervals, page completion intervals, and bandwidth consumption. The process is applied separately to the attack corpus and normal corpus separately and input dataset is converted from the request level to flow level where flow is defined as absolute time interval.

The absolute time intervals (ati) of attack and normal is considered for training, the collection of absolute time intervals (ati) are given as the input for the ensemble of classifiers for defining classifier pool for attack and normal independently. In the testing phase, the input corpus is again converted into absolute time interval (ati) that are validated through ensembles of classifiers. The Adaboost ensemble classifier with different classification algorithms in each level is used to validate the testing corpus as attack or normal.

3.1 The Absolute Time Interval (Ati) Is Defined Using the Following Parameters

- *Collection of Session begin intervals (CSBI)*: This parameter describes the time gap between begin times of the continuous two sessions in the absolute time interval.
- *Collection of Session completion intervals (CSCI)*: This parameter describes the time gap between end times of the continuous two sessions in the absolute time interval.
- *Collection of Page access begin intervals (CPBI)*: This parameter describes the time gap between the begin time of the page access requests in sequence in the absolute time intervals.

- *Collection of Page access completion interval (CPCI)*: This parameter describes the time gap between the completion time of the page access requests in sequence in the absolute time intervals.
- *Bandwidth consumption of Session (SBC)*: This parameter describes the bandwidth consumed by all the requests in each session of absolute time interval.

3.2 Feature Extraction from Dataset

Collection of Session begin intervals (CSBI) are defined as a set *sbi* (*Ci*) of size $|C_i| - 1$ related to specific cluster C_i contains the collection of absolute time interval (ati) $|C_i|$. The set *sbi*(*Ci*) of CSBI of the cluster C_i shown as:

$$\underset{j=1}{\overset{|C_i|-1}{\forall}} \left\{ sbi(C_i) \leftarrow \left(bt(s_{j+1}) - bt(s_j) \right) \right\}$$

Collection of Session completion intervals (CSCI) are defined as a set *sci* (*Ci*) of size $|C_i| - 1$ related to a specific cluster C_i includes the sessions of count $|C_i|$. The set *sci*(*Ci*) of CSCI of the cluster C_i is defined as:

$$\underset{j=1}{\overset{|C_i|-1}{\forall}} \left\{ sci(C_i) \leftarrow \left(abs\left(et(s_{j+1}) - et(s_j) \right) \right) \right\}$$

Collection of Page access begin intervals (CPBI) is stated as set $pbi(C_i)$ of size $|P(C_i)| - 1$ related to the collection of pages $P(C_i)$ which includes the pages in increasing order of session begin times. Let $|P(C_i)|$ represent the amount of pages available in every cluster C_i. The set $pbi(C_i)$ of CPBI of cluster C_i is defined as:

$$\underset{j=1}{\overset{|P(C_i)|-1}{\forall}} \left\{ pbi(C_i) \leftarrow \left(bt(p_{j+1}) - bt(p_j) \right) \right\}$$

Collection of Page access completion intervals (CPCI) is represented as a set $pci(C_i)$ of size $|P(C_i)| - 1$ related to the collection of pages $P(C_i)$ which includes the pages in increasing order of session end times. Let $|P(C_i)|$ represent the amount of pages available in every cluster C_i. The set $pci(C_i)$ of CPCI of cluster C_i is defined as:

$$\underset{j=1}{\overset{|P(C_i)|-1}{\forall}} \left\{ pci(C_i) \leftarrow \left(abs\left(et(p_{j+1}) - bt(p_j) \right) \right) \right\}$$

Bandwidth consumption of Session (SBC) related to a cluster C_i are defined as a set $bwc(C_i)$ of size $|C_i|$, Here $|C_i|$ defines the collection of sessions defined in cluster C_i. The amount of the bandwidth consumed by an individual request defined

in cluster refers to the bandwidth in use. The set $bwc(C_i)$ of bandwidth occupied by every session in cluster C_i is shown as follows:

Step 1. $\underset{j=1}{\overset{|C_i|}{\forall}} \left\{ s_j \exists s_j \in C_i \right\}$ Begin

Step 2. $bwc\ (C_i) \leftarrow \sum_{k=1}^{|s_j|} \left\{ bw(p_k) \exists p_k \in s_j \right\}$ // Total bandwidth consumed $bwc\ (p_k)$ by each page p_k in session is moved to the set $bwc\ (C_i)$

Step 3. End //of Step 1

3.3 Source Cluster Selection for Drift Detection

Later the process of absolute time intervals (ati) grouping by their distribution similarity, the proposed model selects the cluster of absolute time intervals for training. Further, the selected clusters are used for training. The formulation of the cluster selection is as follows:

Let a set $CG = \left\{ cg_1, cg_2 \ldots, cg_{|CG|} \right\}$ be the clusters defined and each cluster $\{cg_i \exists cg_i \in CG \wedge 1 \leq i \leq |CG|\}$ represents a set of absolute time intervals(ati) from each cluster-group which is depicted as follows:

Step 1. $\underset{i=1}{\overset{|CG|}{\forall}} \{cg_i \exists cg_i \in CG \wedge 1 \leq i \leq |CG|\}$ Begin // for each cluster-group cg_i depicted in set CG

Step 2. $csm = 0$

Step 3. $\underset{j=1}{\overset{|cg_i|}{\forall}} \left\{ c_j \exists c_j \in cg_i \wedge 1 \leq j \leq |cg_i| \right\}$ Begin

Step 4. *if* $\left(csm < |c_j| \right)$ // if the number of ati $|c_j|$ in cluster c_j is greater than the value of *csm*

Step 5. $sc(cg_i) = c_j$ // selecting the cluster as source cluster of the cluster-group c_j, since it is having the maximum number of sessions than any of the clusters $c_1\ to\ c_{j-1}$ in cluster-group cg_i selected.

Step 6. $csm = |c_j|$ // considering the number of sessions $|c_j|$ in present cluster c_j as max sessions *csm* of the source cluster $sc(cg_i)$ of the cluster-group cg_i selected.

Step 7. End //of Step 4

Step 8. End //of Step 3

Step 9. End //of Step 1

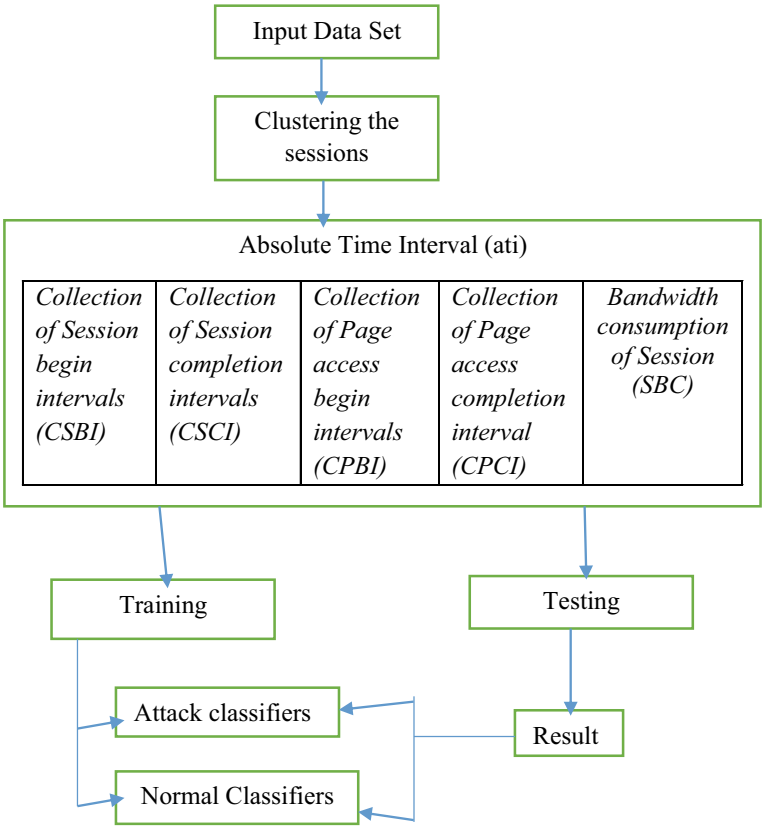

Fig. 1 Process diagram of the proposed model

3.4 Detection Model

As it is mentioned above, the large amount of data delivered during attack is the typical feature of the DDoS attack. The higher base number it is, the little—inaccuracy may lead to large error count, which is still the security problem that needs to be solved. In the detection part, the ensemble training is applied, which votes among the Bagging model, the boosting model, and the meta classifier. Bagging and Adaboost are both the improved model for the weak classifier with the sample training. In order to obtain the stable result of the DDoS detection, two ensemble models and the base classifier are combined.

3.4.1 Bagging

Bagging is a resample mode to training the weak classify which is present as Fig. 1. Sampling the k instances of the data with replacement is the key feature in Bagging model. After nth resample, n sub-datasets are selected as the figure shows. The n independent sub-datasets are trained to predict each own result. The voting procedure is conducted finally to obtain better results [7].

In this way, a new sample that represents the distribution of the original sample is rebuilt with few sample data. That means only few training datasets can also get high result of classification [8]. However, some training samples may be repeated or absent several times in a training session. Because the weight of each classifier is equal, the same mistake may be made in different classifiers. The accuracy of the result normally will increase with the number of resamples. But it may decrease when the resample times to some extent leads to an overfitting result.

3.5 Meta Classifier and Choose Reason

In order to verify the availability of the proposed ensemble framework, more than one classifier are designed to apply in the experiment. Two base classifiers Naive Bayes and J48 are implemented in the proposed design, respectively.

As for Naive Bayes, detecting the abnormal events depends on the probability of the different events [9]. It is used to predict the event as normal or attack by calculating the posterior probability that the event is an attack under the known features' probabilities [10]. That makes the detection rate of DDoS unsteady because the DDoS attack is more complex, which not only combines the different types but also contains different level attacks such as high speed and low speed. J48 predicts the event as normal or attack by calculating the entropy of every feature and divides the different groups by comparing the Information gain of each feature one by one until it identifies the event finally [11]. While J48 usually got the higher detection [12], however, the result may be limited in the size of the dataset and the number of attributes because of the large computing cost in training procedure. For this reason the J48 detection is also unstable with the volume of the DDoS attack dataset having increased. For the large volume of data that needs to be processed in DDoS attack and the various types of the DDoS attack launched nowadays, the two above base classifiers are applied respectively in the proposed voting model which combined the Bagging, Adaboost model, and the base classifier itself. Using the same base classifier separately, mainly because voting among the different detection methods usually lead to the middle result while voting for the different improved versions of one method can get a complementary result by the different sample methods and weighted result.

4 Experiment Configuration

Since the two parts feature selection and detection model were applied in this paper, the results also contained two parts the result of attributes' selection and the final result of detection. The DDoS attack dataset used for the experiment was NSL-KDD dataset which includes 41 attributes as given in Table 1. AS DDoS attack was featured for the high-volume data, detecting every instance with 41 attributes was time-consuming and the computing cost was high. The experiment was conducted on WEKA [13] which is an open platform for data mining. The parameter for ranking and searching is default.

Table 1 The original attributes of the NSL-KDD dataset

Attributes	Attributes
1. duration	22. is_guest_login
2. protocol_type	23. count
3. service	24. srv_count
4. flag	25. serror_rate
5. src_bytes	26. srv_serror_rate
6. dst_bytes	27. rerror_rate
7. land	28. srv_rerror_rate
8. wrong_fragment	29. same_srv_rate
9. urgent	30. diff_srv_rate
10. hot	31. srv_diff_host_rate
11. num_failed_logins	32. dst_host_count
12. logged_in	33. dst_host_srv_count
13. num_compromised	34. dst_host_same_srv_rate
14. root_shell	35. dst_host_diff_srv_rate
15. su_attempted	36. st_host_same_src_port_rate
16. num_root	37. dst_host_srv_diff_host_rate
17. num_file_creations	38. dst_host_serror_rate
18. num_shellsl	39. dst_host_srv_serror_rate
19. num_access_files	40. dst_host_rerror_rate
20. num_outbound_cmds	41. dst_host_srv_rerror_rate
21. is_host_login	

4.1 The Detection Result of Different Data Mining Procedures

In order to present the accuracy of the detection the confusion matrix was calculated. Confusion matrix is one of the most important metrics to evaluate the effectiveness of the attack detection, especially for the multiple attacks such as the sophisticated DDoS attack which contains more than one type of DDoS attack. To evaluate whether the result of the detection is reliable the confusion matrix is the key trait to compare.

It can be seen that True Positive (TP) is the number that attacks were correctly detected, and TN was the number that normal events correctly detected.

While False Negative (FN) meant that the attack instance was regarded as inaccurate, and FP represented the normal event that was regarded as attack. The detailed performances of the two base classifiers and the final result of the proposed model were compared in Table 2. In Table 2, eight parameters of performance are shown. Naive Bayes represented using Naive Bayes as the meta classifier of the RSV model and the same with J48. The As FP Rate was a metric of error rate, the less it was, the high performance the result was. As for other 5 parameters, the higher they were, the high performance the result was. It can be seen from Table 2, each FP Rate of the RSV models were decreased whether using the Naive Bayes or J48 as the base classifier. And every parameter of the two RSV-meta detection was better than base classifiers both Naïve Bayes. It proved that the performance of the RSV detection framework was all-round improved rather than just some aspects.

5 Conclusion

This paper contributes to how the DDoS attack id detected at flow level rather than the request level. From the contemporary literature researchers proposed many techniques to detect and defend the DDoS attacks particularly Application layer DDoS attacks, but nobody has addressed the detection in flow level. The detection accuracy and time is minimized in flow level attack detection rather than request level or session level. In this paper flow is defined with five attributes session begin intervals, session completion intervals, page access begin intervals, page completion intervals, and bandwidth consumption. The Input corpus is converted in terms of absolute time intervals which is known as flow. The ensemble classifiers are used to define multiple classifiers based on the diversity of the traffic, which increases the attack detection accuracy and minimizes the false alarms. In this paper Adaboost is used with different classifiers and validated that the detection accuracy is improved over the traditional and normal request level detection approaches. The overall process is experimented with KDD 99 cup dataset.

Table 2 The performance between the two meta classifiers and the RSV models

Classifier method	TP rate	FP rate	Precision	Recall	F-Measure	MCC	ROC area	PRC area	Class
Ada Boost with J48	0.967	0.03	0.961	0.967	0.964	0.936	0.995	0.994	Attack
	0.97	0.033	0.974	0.97	0.972	0.936	0.995	0.997	Normal
	0.969	0.032	0.969	0.969	0.969	0.936	0.995	0.995	Weighted Avg.
Ada Boost with Naïve Bayes	0.954	0.02	0.973	0.954	0.964	0.936	0.99	0.986	Attack
	0.98	0.046	0.965	0.98	0.972	0.936	0.99	0.992	Normal
	0.969	0.035	0.969	0.969	0.969	0.936	0.99	0.99	Weighted Avg.
Ada Boost with Random forest	0.98	0.03	0.962	0.98	0.971	0.948	0.995	0.99	Attack
	0.97	0.02	0.985	0.97	0.977	0.948	0.997	0.998	Normal
	0.974	0.024	0.974	0.974	0.974	0.948	0.996	0.994	weighted Avg.

References

1. Bhuyan M H, Bhattacharyya D K, Kalita J K (2015) An empirical evaluation of Information metrics for low-rate and high-rate DDoS attack detection. Pattern Recognit Lett 51(C):1–7. https://www.arbornetworks.com/
2. Bhuyan MH, Bhattacharyya DK, Kalita JK (2015) Towards generating real-life datasets for network intrusion detection. Int J Netw Secur 17(6):683–701
3. Alkasassbeh M, Al-Naymat G, Hassanat A et al (2016) Dataset-detecting distributed denial of service attacks using data mining techniques. Int J Adv Comput Sci Appl 7(1):1–10
4. Osanaiye O, Cai H, Choo KKR et al (2016) Ensemble-based multi-filter feature selection method for DdoS detection in cloud computing. Eurasip J Wirel Commun Netw 2016(1):1–10
5. Jia B, Huang X, Liu R et al (2017) A DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning. J Electr Comput Eng 2017(2):1–9
6. Moustafa N, Slay J (2017) The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. In: International workshop on building analysis datasets and gathering experience returns for security. IEEE
7. Breiman Leo (1996) Bagging predictors. Mach Learn 24(2):123–140
8. Tumer K, Ghosh J (2015) Error correlation and error reduction in ensemble classifiers 8(3–4):385–404
9. John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: Eleventh conference on uncertainty in artificial intelligence, San Mateo, pp. 338–345
10. Fouladi RF, Kayatas CE, Anarim E (2016) Frequency based DDoS attack detection approach using naive Bayes classification. In: International conference on telecommunications and signal processing. IEEE
11. Quinlan R (1993). C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Mateo, CA
12. Patel J, Panchal K (2015) Effective intrusion detection system using data mining technique
13. Hall M, Frank E, Holmes G et al (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18