International Journal of
Science and Engineering Investigations vol. 7, issue 72, January 2018
ISSN: 2251-8843

Using Genetic Algorithm to Improve Bernoulli Naïve Bayes Algorithm in Order to Detect DDoS Attacks in Cloud Computing Platform

Ali Mahmodi Derakhsh[1], Parisa Daneshjoo[2], Changiz Delara[3]
[1,2,3]Department of Computer Engineering, West Tehran Branch, Islamic Azad University, Tehran, Iran
([1]Mahmodi.Ali@wtiau.ac.ir, [2]Daneshjoo.p@wtiau.ac.ir, [3]Delara.c@wtiau.ac.ir)

Abstract-Devices such as routers, switches or firewalls are the most vital connections in communication network among physical machines in a cloud computing environment. In the absence of security on the network, intruders are allowed to access the equipment and configure it in the way they want to. Hence, a method suggested to deal with denial-of-service (DoS) attacks in the cloud computing platform is one of the essential and most important security issues in this area. This study tends to provide a smart method based on Bernoulli naïve bayes algorithm focusing on genetic algorithm for detecting DoS attacks. Through different network streams, network streams which trigger DoS and DDoS attacks are very important.

The main idea of this study is to use Bernoulli naïve bayes algorithm to identify DoS attacks, which is the main reason for optimizing this algorithm using genetic algorithm. In this method, an optimal subset of the set of features is extracted using genetic algorithm, and this optimal subset is used for Bernoulli naïve bayes learning. Results of the experiments carried out and comparison of the suggested method with other methods indicate proper accuracy and operation of the suggested method.

Keywords- *Cloud Computing, Network Security, Denial-of-Service Attacks, Genetic Algorithm, Bernoulli Naïve Bayes Algorithm*

I. INTRODUCTION

To provide security on a network, one of the most critical and most dangerous steps is to provide security of access and control of network equipment. Devices such as routers, switches or firewalls are the most vital connections in communication network among physical machines in a cloud computing environment. Equipment security is particularly important for two reasons [1]. The lack of security of equipment on the network allows intruders to access the equipment and configure it in the way they want to. In this way, any intrusion or theft of information or any other damage to cloud computing environment will be possible by the intruder. To avoid the risks of denial of service (DoS), security of equipment is required on the network. Intruders can exploit services on the network by these attacks. Hence, a method suggested dealing with DoS attacks in cloud computing platform is one of the essential and most important security issues in this area. This study tends to provide a smart method based on Bernoulli naïve bayes algorithm focusing on genetic algorithm for detecting DoS attacks. The main idea of this study is to use Bernoulli naïve bayes algorithm to identify DoS attacks, which is the main reason for optimizing these algorithm using genetic algorithm as a brand new method. [2] Explores various methods of machine learning and data mining used to identify DoS attacks. Methods such as support vector machine, Bayesian networks, and decision tree are the most important algorithm evaluated in this review. This review is a comprehensive, complete and new resource for familiarizing with methods of dealing with DoS attacks using data mining techniques. In [3], DoS attacks are identified using a classifier. The classifier used in this method is a support vector machine, which is one of the most efficient classifiers. The main point of this study is to use this classifier to identify DoS attacks in a SDN-equipped network. [4] Develops a framework for identifying DoS attacks using genetic algorithm. Given that genetic algorithm are used to solve optimization problems, detection of DoS attacks is mapped to an optimization problem and genetic algorithm are used to solve the problem. The experiments show that the suggested method is well accurate. In [5], feature selection method is used to find optimal features for intrusion detection and detection of DDoS attacks. The main feature of this method is to use filters to select optimal feature set. Finally, a support vector machine algorithm is used to detect intrusion. The results indicate that this method is very accurate. In [6], filters are used to reduce feature and identify the optimal feature set. The difference between this method and the preceding paragraph is that: 1) this method uses multiple filters, and 2) the suggested method is real-time and can be run on stream data. This method focuses on intrusion detection in cloud computing environment. In [7], the main focus is on feature selection and detection. For this purpose, fuzzy entropy method is used. In addition, ant colony method is used to optimize and select optimal features. Another important feature of this method is its real-time nature, which distinguishes it from other methods. In sum, this method has good innovations and very satisfactory results.

II. The Suggested Method

As discussed earlier, the method presented in this study is based on machine learning and optimization of genetic algorithm. In this study, the Bernoulli naïve bayes classifier is used. The goal is to optimize the above classifier by genetic algorithm, and finally, compare accuracy of DDoS attack detection by these algorithms and select the best algorithm.

A. General Trend of Genetic Algorithm

Figure 1 shows a standard genetic algorithm and Figure 2 shows flowchart of genetic algorithm. Before a genetic algorithm can be run, the problem must be encoded (or displayed) properly. Moreover, a fitness function must be invented to assign a value to any encoded solution. During the run, parents are selected for reproduction and combined together by crossover and mutation operators to generate new offspring. This process is repeated several times until the next generation of the population is generated. Then this population is examined and the process ends if the convergence criteria are met.

1) Tournament Selection Method

The main idea of selection methods is that better people are preferred over worse people, which is defined by the fitness function f. Several selection methods have been suggested for use in genetic algorithm. One of the good features of selection methods is that these methods are independent of presentation of population and only fitness values of people are considered.

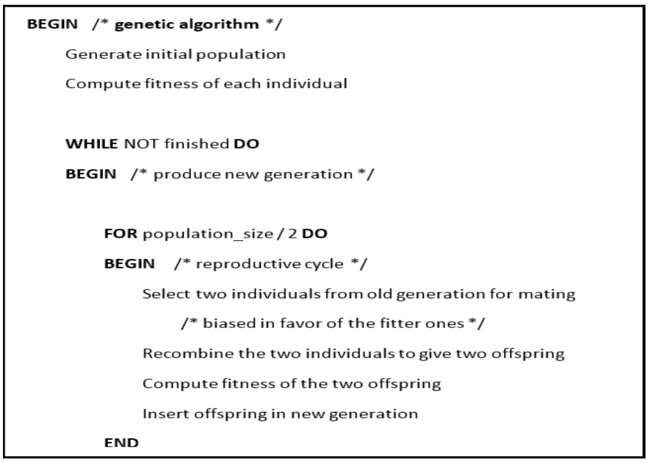
Tournament selection is similar to selecting a rating based on selection pressure, but it is more efficient for calculations and more suitable for parallel implementations. In this method, two people of the population are randomly selected. Then, a random number, r, is chosen between zero and one. If r<k (where, k is a parameter, for example, 0.75), the person is more fitted; otherwise, the less fitted person is selected as the parent. These two are then returned to the primary population and are again included in the selection process.

2) Termination Criteria

In evolutionary algorithm, the program is often run for predetermined generations; however, another condition has been considered for terminating genetic algorithm by Grefenstette [8], which is bit diversity within the population. This criterion represents the degree of convergence of population code. If the code of an element is of a length of one bit and is represented as $\bar{a}_i(a_{il}، ... ،a_{it})$ and $\{1، ... ، \mu\}$ (where, μ is the number of members of the population), bit diversity of the population P, b (P), is defined as:

$$b(p) = \frac{1}{\mu.1}\sum_{j=1}^{1} Max\{\sum_{i=1}^{\mu}(1-a)، \sum_{i=1}^{\mu} a_{ij}\} \in [0.5، 1.0] \quad (1)$$

Whatever the size of b is larger, bit diversity will be less in the population. Particularly, if b (P) = 1, the code of all members of the population is the same. Termination criterion is defined as $b(p) > b_{max}$, where b_{ma} is usually ≈0.95.



```
BEGIN   /* genetic algorithm */
    Generate initial population
    Compute fitness of each individual

    WHILE NOT finished DO
    BEGIN   /* produce new generation */

        FOR population_size / 2 DO
        BEGIN   /* reproductive cycle */
            Select two individuals from old generation for mating
                /* biased in favor of the fitter ones */
            Recombine the two individuals to give two offspring
            Compute fitness of the two offspring
            Insert offspring in new generation
        END
```

Figure 1. Standard genetic algorithm

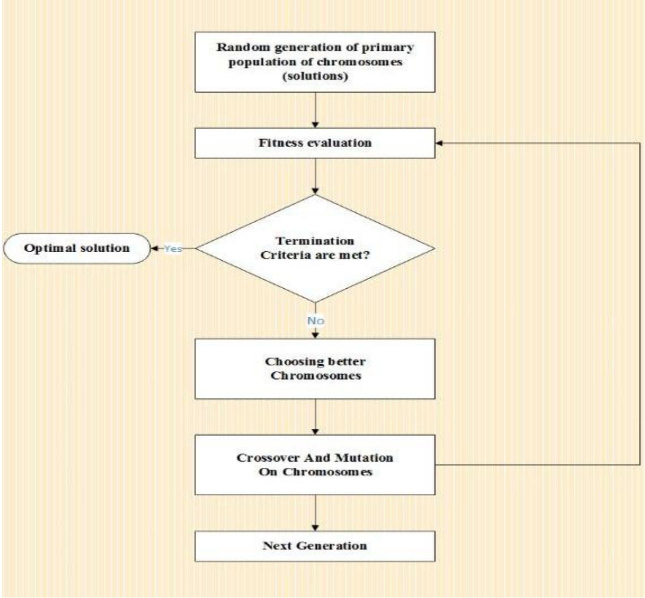

Random generation of primary population of chromosomes (solutions)
Fitness evaluation
Termination Criteria are met?
Optimal solution
Yes
No
Choosing better Chromosomes
Crossover And Mutation On Chromosomes
Next Generation

Figure 2. flowchart of genetic algorithm

B. Bernoulli Naïve Bayes

Bayes approach is a method to classify events based on occurrence probability or not happening. naïve bayes probability shows proper results using native characteristics when it receives primitive practice. Learning technique in naïve is a learning style with observer.

For example a fruit might be orange. If its color is orange, its round and has about 10cm of radius. If these probabilities be dependent in a correct way, naïve bayes will act accurately in recognizing if it's orange or not.

There are many applications that estimate naïve bayes parameters, thus people can utilize this possibility to solve their problems without knowing about bayes theory. Using design problems and assumptions that exist in bayes, this method is suitable for problem classification in the real world.

While the data has a multi-variable Bernoulli distribution, Bernoulli naïve bayes is utilized. in this situation samples will be considered as binary variables. xi sample will be given, classification decision making rule will be defined as follow[9]:

$$p(x_i|Y) = p(i|Y)x_i \times (1 - p(i|Y))(1 - x_i) \qquad (2)$$

The goal pursued in this study is to use genetic algorithm to improve classifier. The main purpose of genetic algorithm is feature selection given that DDoS dataset generally has a large number of features, a proper method used for feature selection also has a significant effect on performance of the Bernoulli naïve bayes classifier. The general procedure for using genetic algorithm to select a feature of the feature set is described below.

Primary population is randomly generated. Each sample of the population contains n genes, which is equal to the number of features in the dataset. In other words, each gene determines whether the corresponding feature is used in the model construction; if yes, its value is 1 and otherwise, zero. As a result, each sample in the population represents the selection for the existing features. For each sample in the current population, the corresponding model is developed.

Once the corresponding naïve bayes model is developed, this model is evaluated by a validation data set and its classification error rate is extracted. The naïve bayes, which has a lower classification error rate, is a better sample.

When evaluation function or the same classification error rate is computed for all samples of the population, genetic algorithm generates the next generation as follows:

1. Selection of samples for the next generation using rank selection method described below
2. Two point crossover is used to generate offspring (Figure 3), which is calculated as follows: In this method, two points are selected in the parent's chromosome, and everything between these two points is moved to generate new ones.

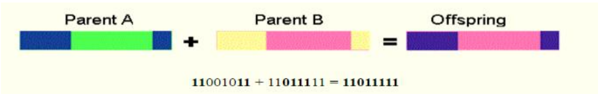


Figure 3. **T**wo-point crossover [10]

3. In each offspring, bit level mutation is used for mutation. In this method, a bit is selected randomly and its value is moved, that is, it becomes one if it is zero, and it becomes zero if it is one. The probability of selecting each bit for mutation is $\frac{1}{l}$, where l is length of the chromosome. Two best samples are kept and the entire current population is replaced by offspring.

These steps are repeated frequently until maximum number of repetitions (1000 times) is run.

The main thing that exists in genetic algorithm and can have a significant effect on its performance and prevent rapid convergence is selection of better chromosomes. In the following, some typical selection methods used in genetic algorithm are explained. Suppose, the following roulette wheel (Figure 4) is made for all chromosomes, so that each chromosome with higher fitness functions value occupies a larger area.

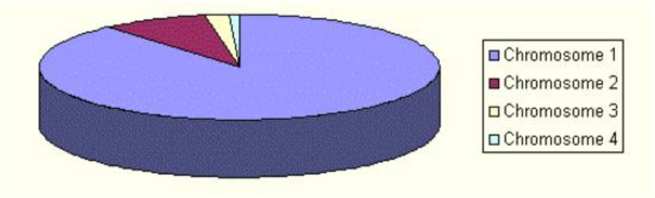


Figure 4. **R**oulette wheel of chromosomes [10]

In the roulette wheel routine, the chromosome with the highest fitness function value has a greater chance of selection, which is described below:

1. The sum of all fitness function values is computed for each chromosome and placed in the variable S.
2. A random value, r, is selected between (0, S).
3. On population of chromosomes and fitness function values, it is passed through zero to S until the current value is larger than r; in this case, the operation stops and the corresponding chromosome are returned.

In this case, chromosomes with higher fitness values obviously have a higher chance of selecting; however, if a chromosome has a very high fitness value (e.g., 90% of total values of fitness functions), there will be little chance of selecting other chromosomes. In rank selection to solve this problem, a rank value is assigned to each chromosome; 1 for the worst chromosome, 2 for the second worst and, finally, N for the best chromosome. In this case, all chromosomes have the same chances to be selected (Figure 5); however, the main problem of this method is its slow convergence, since no difference is made between the best chromosome and other chromosomes.

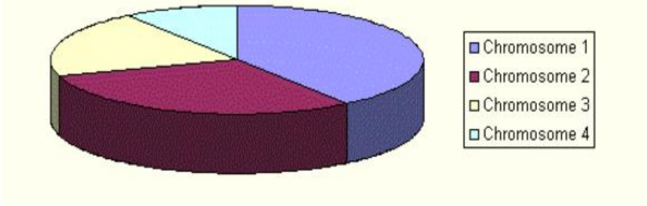


Figure 5. Roulette wheel of chromosomes for rank selection

Obviously, each of the above methods has some kind of deficiency or defect in performance. In this study, the method presented in [11] is used which showed a good performance in experiments and analyses performed.

Steps of this algorithm include:

1. Population of chromosomes is sorted according to their fitness function value.
2. Population is divided into two parts. The part with higher fitness values (HF) and the part with lower fitness values (LF) are determined, where population breakdown is also expressed by parameter m in percentage.
3. Always the first parent (P1) is selected from HF and the second parent is selected from LF.
4. Offspring P1 and P2 are calculated.
5. Offspring is added to the end of population.
6. The population is sorted again based on fitness function value and end chromosomes are removed to keep the population length constant.

C. Analysis of the Suggested Method

Genetic algorithm is used to identify and select an effective subset of features for classification. In other words, the features which have the greatest effect on data separation are selected among the primary features and then classification is done based on these features. To illustrate the effect of features, support vector decomposition [12] algorithm is used to calculate eigenvalue for each feature in the dataset. Higher eigenvalues indicate higher effect of the corresponding feature in data separation or definition. This study uses NSL-KDD 2013 [13] dataset, which is the extended dataset of KDDCup99 [14] and includes 41 features. In the pre-processing process, features which do not have numeric values are broken into several numerical features, so that these features can be used in Bernoulli naïve bayes algorithm. Therefore, the number of features will be considerably higher than 41 features after pre-processing. Figure 6 shows eigenvalues for features of this dataset in a descending order. As shown in this figure, approximately 20 out of 150 features (after performing the necessary pre-processing) of this dataset have eigenvalue larger than zero, indicating that a large amount of features (about 130 features) have no effect in defining or distributing data. Therefore, this study used genetic algorithm to identify highly effective features in classifier performance.

Next, the effect of genetic algorithm is examined on result of the considered classifier. Figure 7 shows the difference in classifier efficiency with and without genetic algorithm in baseline for Bernoulli naïve bayes classifier.

To evaluate classifier performance, the parameter, receiver operating characteristic (ROC) [15], is used which is a diagram in terms of false positive ratio (fallout) and true positive rate (sensitivity). The greater the area below ROC diagram, the more accurate the corresponding classifier produces the correct answer; in other words, the higher the area below the diagram, the better the efficiency of the corresponding classifier. As shown in the figure, it is considered in the Bernoulli naïve bayes classifier that better performance will occur if genetic algorithm is applied. According to explanations presented, efficiency of this method is clear in improving the accuracy of Bernoulli naïve bayes classifier.

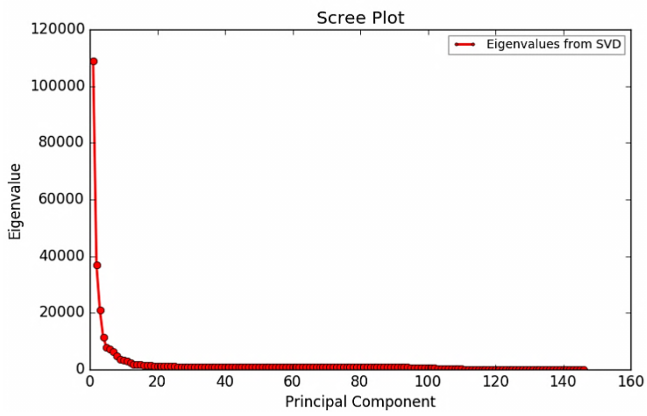


Figure 6. **E**igenvalues for features of dataset NSL-KDD

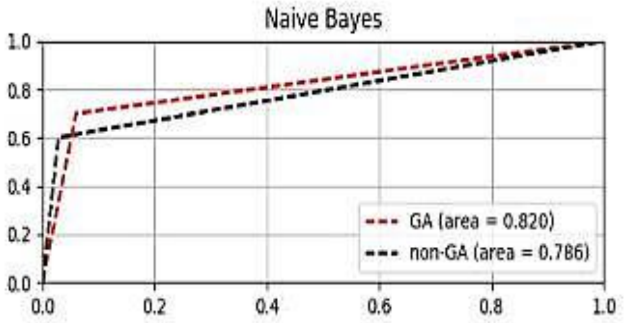


Figure 7. **C**omparison of efficiency of bernoulli naïve bayes with and without genetic algorithm in baseline

III. RESULTS

A. Dataset

The dataset used in this study to implement the suggested method is NSL-KDD dataset [13]. This dataset contains three main files:

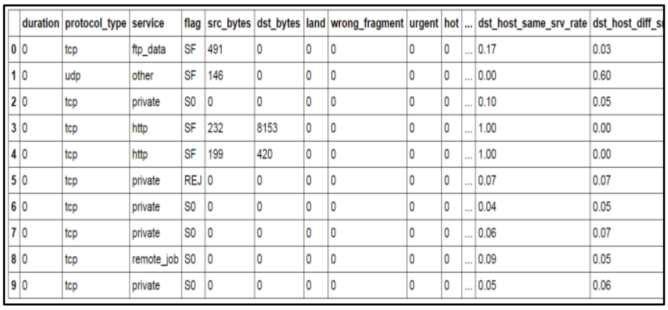
1. Training dataset contains 125973 data, each of which contains 41 features.
2. The first test dataset contains 22544 data.
3. The second test dataset contains 11850 data

Figure 8 shows an overview of this dataset. Given that the method suggested in this study is used to identify DoS attacks in cloud computing platform, the algorithm was compared independent of requirements of cloud computing environment with common methods to ensure accuracy, efficiency and performance of the algorithm compared with basic methods. In other words, the condition or the hypothesis which damages general issues of DoS attacks is avoided.

B. *Dataset Preprocessing*

The above dataset requires a pre-processing process to run Bernoulli naïve bayes algorithm and genetic algorithm in the following phases:

1. Removing null values: To remove null values, these values were replaced by mean values of the same feature in the dataset.
2. Normalizing features: Due to the fact that the range of feature values varies and they differ greatly, the range of these values must be homogeneous to improve the performance of machine learning algorithm, which was also applied to the dataset. Z-score [16] was used to normalize the features.



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_s
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	0.17	0.03
1	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60
2	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05
3	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00
4	0	tcp	http	SF	199	420	0	0	0	0	...	1.00	0.00
5	0	tcp	private	REJ	0	0	0	0	0	0	...	0.07	0.07
6	0	tcp	private	S0	0	0	0	0	0	0	...	0.04	0.05
7	0	tcp	private	S0	0	0	0	0	0	0	...	0.06	0.07
8	0	tcp	remote_job	S0	0	0	0	0	0	0	...	0.09	0.05
9	0	tcp	private	S0	0	0	0	0	0	0	...	0.05	0.06

Figure 8. A view of the dataset NSL-KDD

C. *Implementation of the Suggested Method*

To implement the suggested method, Python programming language version 3.5 was used. The hardware specifications on which the algorithm was run are listed in Table 1. For implementation of this method, libraries of machine learning algorithm, genetic algorithm and mathematical matrix calculations were used in Python. The list of these libraries, along with their descriptions, is presented in Table 2.

TABLE I. SPECIFICATIONS OF IMPLEMENTATION HARDWARE

Operating system	Windows 7, 64-bit
Programming language	Python version 3.5.1
Text editor	JetBrains Pycharms 2016
Processor	Core i7-4510U
Main memory	8 GB

TABLE II. THE USED LIBRARIES

Library	Application
Scikit-learn	Machine learning algorithm
deap	Genetic Algorithm
numpy	Mathematic matrix calculations

D. *Implementation Parameters*

This section gives parameters considered for genetic algorithm during run and implementation. Table 3 shows these parameters and their values.

TABLE III. VALUES OF GENETIC ALGORITHM PARAMETERS

Parameter	Value
Number of species	1000
Number of steps of run	10000
The probability of combining two species by crossover.	0.5
Mutation probability for any species	0.2

E. *Evaluation Parameters*

The following parameters are used to evaluate the suggested method. Confusion matrix shows how classification algorithm performs according to input dataset in terms of various classes of classification problem [17].

TABLE IV. CONFUSION MATRIX

	Predicted		
Actual		Positive	Negative
	Positive	True Positives(TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Each of the matrix elements is as follows:

TN: represents the number of records whose actual class is negative and classification algorithm correctly identifies it as negative.

TP: represents the number of records whose actual class is positive and classification algorithm correctly identifies it as positive.

FP: represents the number of records whose actual class is negative and classification algorithm incorrectly identifies it as positive.

FN: represents the number of records whose actual class is positive and classification algorithm incorrectly identifies it as negative.

In the following, various criteria and parameters required to evaluate the suggested method and compare it with other methods are defined. All of the following definitions can be deduced from confusion matrix.

$$Precision = \frac{tp}{tp+fp} \tag{3}$$

$$Recall = \frac{tp}{tp+fn} \tag{4}$$

$$True\ negative\ rate = \frac{tn}{tn+fp} \tag{5}$$

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{6}$$

$$F - Measure = 2 . \frac{precision\ .recall}{precision+recall} \tag{7}$$

F. Evaluation of the Suggested Method

This study used Bernoulli naïve bayes algorithm to identify DDoS attacks. The main innovation of this study is to use genetic algorithm to select an optimal subset of dataset features to improve performance of the Bernoulli naïve bayes algorithm. Table 5 gives the results of evaluation of the above algorithm with and without genetic algorithm.

TABLE V. COMPARISON OF DIFFERENT VERSIONS OF THE SUGGESTED METHOD

Method	KDDTest+				KDDTest-21			
	Precision	Recall	F-Measure	Accuracy	Precision	Recall	F-Measure	Accuracy
Naïve Bayes	0.83	0.76	0.76	0.7598	0.82	0.55	0.59	0.5453
Naïve Bayes + GA	0.85	0.82	0.82	0.8244	0.81	0.64	0.68	0.6437

As indicated in the table above, Bernoulli naïve bayes algorithm optimized by genetic algorithm has better performance and efficiency than Bernoulli naïve bayes algorithm. In terms of algorithm runtime under different conditions, a comparison is also made in Table 6. However, note that this time is the time it takes for an algorithm to train, which is an offline time.

TABLE VI. ALGORITHM RUNTIME IN MINUTE

Method	Run Time (min)
Naïve Bayes	3
Naïve Bayes + GA	484

IV. CONCLUSION

This study developed a method based on genetic algorithm to identify DDoS attacks in cloud computing platform. The main approach in this study was to optimize Bernoulli naïve bayes classifier using genetic algorithm. In other words, genetic algorithm were used to find a subset of features which causes most precision in detecting DDoS attacks in Bernoulli naïve bayes algorithm. Finally, the suggested method is run on a valid dataset and the results indicated higher performance and accuracy of the suggested method.

REFERENCES

[1] Wang, B., et al., DDoS attack protection in the era of cloud computing and software-defined networking. Computer Networks, 2015. 81: p. 308-319.

[2] Tama, B.A. and K.-H. Rhee, Data Mining Techniques in DoS/DDoS Attack Detection: A Literature Review. International Information Institute (Tokyo). Information, 2015. 18(8): p. 3739.

[3] Li, X., et al. DDoS Detection in SDN Switches using Support Vector Machine Classifier. in 2015 Joint International Mechanical, Electronic and Information Technology Conference (JIMET-15). 2015. Atlantis Press.

[4] Malhi, A.K. and S. Batra, Genetic-based framework for prevention of masquerade and DDoS attacks in vehicular ad-hocnetworks. Security and Communication Networks, 2016. 9(15): p. 2612-2626.

[5] Ambusaidi, M.A., et al., Building an intrusion detection system using a filter-based feature selection algorithm. IEEE transactions on computers, 2016. 65(10): p. 2986-2998.

[6] Osanaiye, O., et al., Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. EURASIP Journal on Wireless Communications and Networking, 2016. 2016(1): p. 130.

[7] Varma, P.R.K., V.V. Kumari, and S.S. Kumar, Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-time Intrusion Detection System. Procedia Computer Science, 2016. 85: p. 503-510.

[8] Grefenstette, J.J., Optimization of control parameters for genetic algorithm. IEEE Transactions on systems, man, and cybernetics, 1986. 16(1): p. 122-128.

[9] Murphy, K.P., *Naive bayes classifiers.* University of British Columbia, 2006.

[10] Mitchell, M., An introduction to genetic algorithm. 1998: MIT press.

[11] Ali, E. and E. Elamin, A proposed genetic algorithm selection method. 2006.

[12] Pereira, F. and G. Gordon. The support vector decomposition machine. in Proceedings of the 23rd international conference on Machine learning. 2006. ACM.

[13] Revathi, S. and A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. 2013.

[14] Cup, K., Dataset. available at the following website http://kdd. ics. uci. edu/databases/kddcup99/kddcup99. html, 1999. 72.

[15] Hanley, J.A. and B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 1982. 143(1): p. 29-36.

[16] Cheadle, C., et al., Analysis of microarray data using Z score transformation. The Journal of molecular diagnostics, 2003. 5(2): p. 73-

[17] Visa, S. Ramsay, B. Ralescu, A. and VanDerKnaap, E., Confusion Matrix-Based Feature Selection. Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference 2011.

Ali Mahmodi Derakhsh was born in Maragheh, Iran, in Feb 1984. He received the B.S. degree in Software Engineering from the West Tehran Branch, Islamic Azad University (WTIAU), Tehran, Iran in 2011. He is currently MSc. Student in Information Technology Engineering from the West Tehran Branch, Islamic Azad University (WTIAU), Tehran, Iran. Since 2010 he has researched about Data Mining and Machine Learning and published several papers related to this field. Eng. Mahmodi's research interests include machine learning, Security, Genetic algorithm, Cloud computing.

Parisa Daneshjoo was born in Tehran, She received her B.Sc. degree in Computer Software Engineering in 1996, from Islamic Azad University, South Tehran Branch, and Tehran, Iran. She received her M.Sc. degrees in Computer Software Engineering in 2008 from Tarbiat Modares University, Tehran, Iran and received Ph.D. degrees in Software Engineering in 2015 from Islamic Azad University, Science and Research Branch, Tehran, Iran respectively. She held the position of Assistant Professor in Islamic Azad university west Tehran branch (WTIAU). She was Head of Department, Computer engineering in Islamic Azad university west Tehran branch, Tehran, Iran (WTIAU) in 2015-2017.

Changiz Delara was born in Astara, he received his B.Sc. degree in Statistics and Computer Science in 1977 from Ferdowsi University, Mashhad, Iran. He received his M.Sc. degrees in Computer Science in 1980 from The George Washington University, USA and Ph.D. degrees in Intelligent Software Engineering in 1996 from University of Sussex, UK. He is Assistant Professor at west Tehran branch of Islamic Azad University of Iran since 2010.