Towards an Effective Approach to Detection Distributed Denial of Service Attacks Using Genetic Algorithm and Neural Networks for the Internet of Things

Rusul A. Enad
Math. Department, Al-Zahraa University for Women, Iraq,
Karbala, Iraq
rusul.ali@alzahraa.edu.iq

Alyaa M. Al-khafagy
Math. Department, Al-Zahraa University for Women, Iraq,
Karbala, Iraq

Sarah R. Hashim
Math. Department, Al-Zahraa University for Women, Iraq,
Karbala, Iraq

Ihab L. Hussein Alsammak
Ministry of Education, Directorate General of Education of Karbala ehablaith@gmail.com

Abstract

Smart devices are classified into a general class called the Internet of Things. The Internet of Things refers to the interconnection of various entities via the Internet. The security challenge is a major concern for developers and users of the Internet of Things, which must be considered from different perspectives. There are several vulnerabilities in the Internet of Things. In this paper, we focus on distributed denial of service (DDoS) attacks that can block the access of legitimate users and make network resources inaccessible. When a similar attack originates from multiple sources, it is referred to as a DDoS. The objective of this work is to provide an intelligent security solution to detect this type of attack and prevent service disruption on the Internet of Things. The proposed approach to detecting DDoS attacks was evaluated using the MATLAB programming language. The evaluation steps were performed using the criteria of precision, recall, accuracy, and F_Measure in two different scenarios. In the first scenario, the size of the dataset is constant and the accuracy results range from 65% to 99%, but in the second scenario, the size of the training dataset varies and the accuracy results range from 67% to 99%.

Keywords: *Internet of Things; Attack detection; DDoS; Genetic Algorithm; Neural Networks.*
DOI Number: 10.48047/nq.2022.20.19.NQ99169 **NeuroQuantology2022;20(19): 1943-1957**

1. Introduction

The next generation of the Internet of Things (IoT) will impact all aspects of our lives. This technology will enable users to collect, shop, and communicate a wide range of sensitive and personal data anywhere, anytime [1]. By reaching the ultimate goal of the Internet of Things, all entities can exchange information and process data according to predefined designs. In such a system, not only is the environment always available to users, but also all the functions of the devices connected to this environment are always available, and consequently their meaning is clear. In recent years, this trend has intensified in the field of the Internet of Things due to the miniaturization of devices[2], the increase in computing power, and the reduction in energy consumption [3].

One of the main IoT challenges attracting the attention of experts is security. The increase in IoT threats and attacks has highlighted the need for new techniques to detect and defend against attacks. These threats and attacks can be assessed from different perspectives [4] , However, one difficult problem in the field of intrusion detection systems is assessing the progress made in identifying malicious

code[5]. Attacks are divided into two classes: active and passive, and from another perspective, they are divided into two classes: destructive and non-destructive. An attack on the network, regardless of its nature[6], can cause irreparable damage to users, devices, objects, and communications, but one of the most important attacks is the denial of service attack, which disrupts the provision of services and communications within the network, leading to the disruption of the entire network. Traditional solutions in this area have many vulnerabilities. Conventional intrusion detection methods (IDS) are technically very complex and rely on transient methods of trial and error [7].

Research on anomaly detection has shown the benefits of machine learning (ML) in detecting malicious IoT traffic [8]. However, limited efforts have been made to develop IoT-specific ML models for detecting attack traffic. Fortunately, IoT traffic is significantly different from other networked devices (e.g., laptops and smartphones)[9], as IoT devices often communicate with a small number of endpoints (rather than a large number of web servers). Previous research has shown that certain behaviors in IoT (e.g. a limited number of endpoints and regular intervals between incoming packets, as well as regular pings at regular intervals) can contribute to feature selection and increase the accuracy of detecting DDoS attacks in IoT traffic.

The purpose of an intrusion detection system, in addition to preventing attacks, is to identify and detect security vulnerabilities in the system or computer network and report them to the system administrators. Intrusion detection systems are usually used as a supplement to firewalls. The task of intrusion detection systems is to detect any unauthorized use of the system by internal and external users [10].Numerous software and hardware intrusion detection systems have been developed, each with their own advantages and disadvantages[11].

Section 2 provides an overview of the literature. Section 3 presents the proposed method. Section 4 discusses the test results, and the last section presents the conclusion of the paper.

2. Literature review

In [12], a network security tool called AGURI is presented. Using a collection of traffic patterns, this tool can monitor network traffic over a long period of time and detect a denial-of-service attack. In[13], a comparative and sequential method for rapid detection of denial-of-service attacks is presented. This method is based on statistical analysis of information across different network layers and detects sudden changes in traffic. The basic idea of this method is that a denial-of-service attack causes a sudden change in the statistical models of the traffic. Also, in [14], a data structure called "MULTIOPS" is introduced to detect denial-of-service attacks. Based on this data structure, the input data stream into the network is.

An attack warning is issued when an asymmetry between the two structures is detected. Then, [15] network traffic is analyzed in terms of TCP token rates and various contract rates. In this paper, it is shown that these rates change significantly when an attack occurs. Furthermore, a set of state rules is created and evaluated by machine learning algorithms to detect denial of service attacks.

In their book, Butty and Hubaux [16] propose the use of game theory for security problems in wireless networks. Alpcan and Basar [17] address a wide range of security issues based on game theory. In numerous studies, game theory has been used to investigate intrusion detection systems. The logical reason for this is that conventional IDSs are based on decision theory. Moreover, game theory is more suitable for security problems than traditional decision theory. In [16], game theory approaches for IDS are described in terms of different game models. This research [18] compared the use of game theory and decision theory for software configuration challenges. Their study showed that the game-theoretic approach is generally more suitable than the decision-theoretic approach.

In [19], a multilevel dynamic theory was used to study the intrusion detection problem in mobile ad hoc networks. The method proposed in [20] models the policy-based IDS configuration as a dynamic random game. A random game model has also been considered in research [21] for the problem of attack by internal members of the organization. The game

method presented in [22] was proposed to study the challenge of intrusion detection in wireless ad hoc networks.

In [23], the destructive signal problem is considered in a scenario called Rayleigh MIMO Gaussian. The interaction between the destructive signal generator and the transmitter/receiver pair is modelled as a zero-sum game. In this game, the attacker tries to minimize the interaction between the transmitted and the received signal, while the defenders try to maximize it.

In [24], a method based on game theory was proposed for detecting denial-of-service attacks in the Internet of Things. An attacker in the Internet of Things attempts to redesign the home page of a particular server. A random game method is proposed between the network manager and the attacker, where both players choose their actions at each time step and the game is moved to a new state according to the probabilities for the chosen actions. The authors showed through simulations that the game goes through several Nash equilibria.

To more insight, a summary of the latest studies is shown in Table 1 below:

Table 1 **Summary of main methodologies for detecting DDoS attacks in recent studies**

Study	Methodology
[12]	by AGURI tool can monitor network traffic over a long period of time and detect a denial-of-service attack.
[13]	Using a comparative and sequential method, sudden changes in traffic are identified. The main idea is that denial-of-service attacks cause a sudden change in the statistical models of traffic.
[14]	a data structure called MULTIOPS is introduced to detect denial of service attacks.
[15]	Network traffic is analyzed in terms of TCP token rates and various contract rates. A set of state rules is also generated and evaluated by machine learning algorithms to detect denial of service attacks.
[16]	suggest the use of game theory for wireless network security problems.
[17]	Game theory has been used in numerous studies to investigate intrusion detection systems. The logical reason for this is that conventional IDSs are based on decision theory. Moreover, game theory is more suitable for security problems than traditional decision theory.
[18]	the use of game theory and decision theory for software configuration challenges has been compared. Their study has shown that, in general, the game theory approach is more appropriate than the decision theory approach.
[19]	multi-stage dynamic theory has been used to study the problem of intrusion detection in mobile ad-hoc networks.
[20]	models the policy-based IDS configuration as a dynamic random game.
[21]	Using a random game model to solve the problem of attacks by internal members of

the organization.

[22] By using the game method to study the challenge of intrusion detection in wireless ad hoc networks.

[23] the problem of destructive signals is considered in a scenario called Rayleigh MIMO Gaussian.

[24] a method has been proposed to detect attacks of denial of service on the Internet of Things based on game theory.

3. Methods

The approach we propose to detect DDoS attacks consists of two basic phases. In the first phase, the dataset containing the different sessions and their associated features is evaluated using a genetic algorithm-based approach, and redundant features are removed. In the second phase, the optimal dataset is given to the multilayer perceptron network, and the sessions are classified into two classes: healthy and invasive. In the following sections, each of these phases is examined.

3.1. *Phase 1: Reduction of features*

Abbreviations and acronyms are defined when they are used for the first time in the text, even if they have already been defined in the summary. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not need to be defined. Do not use abbreviations in the title or heading unless they are unavoidable.

In the genetic algorithm, the structure of the chromosomes, the type of genes, the type of cross-over, the mutation method, and the fitness function are determined according to the type of application. All symbols used in the equations should be defined in the following text.
For a better insight into the phases of the genetic algorithm, see Figure 1 below:

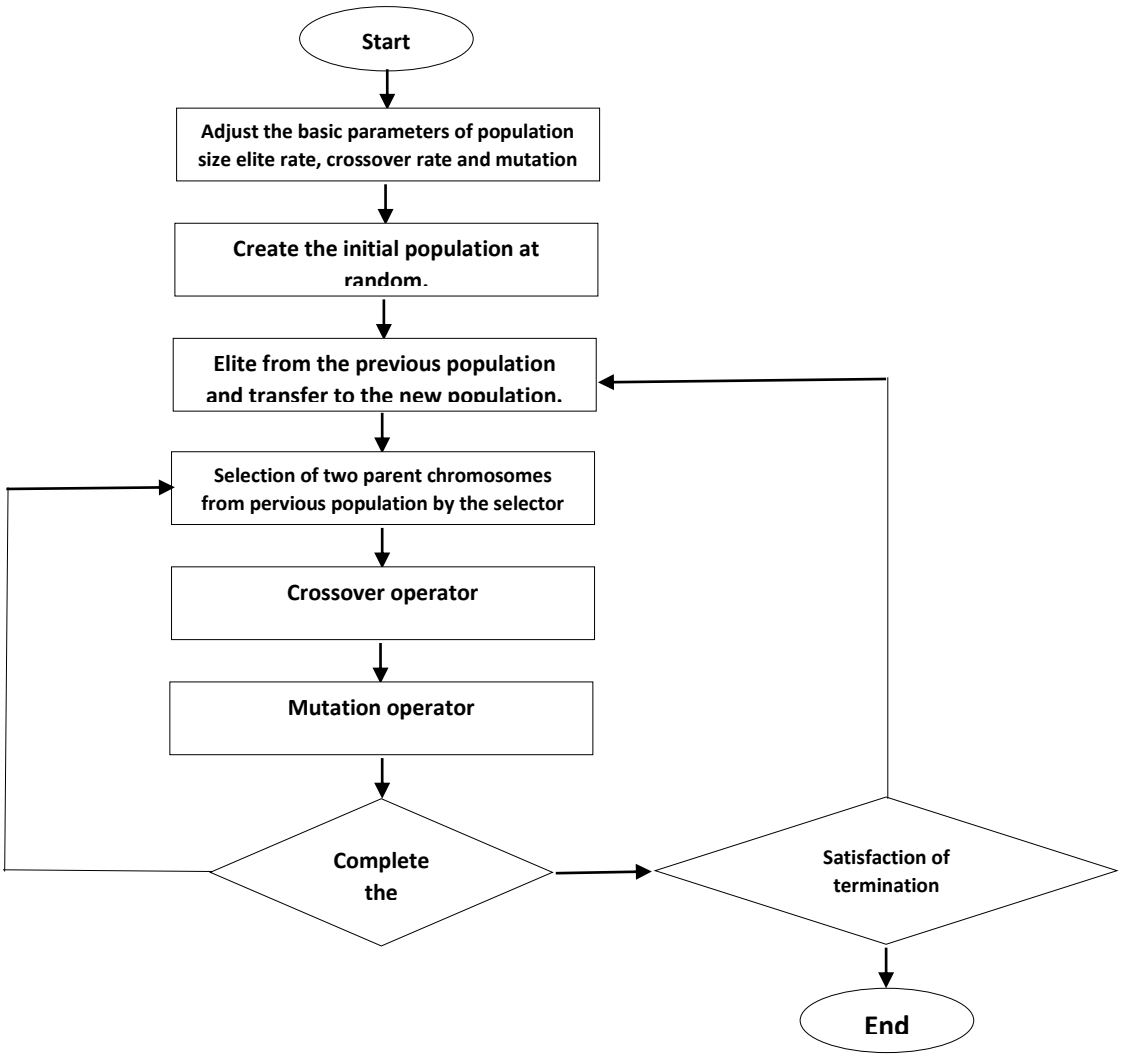


Figure 1 Phases of genetic algorithm.

3.1.1 Chromosome structure

F_1	F_2	F_3	F_4	.....	F_n
1	0	0	1	.....	1

The chromosome structure considered in this study is shown in Figure 2.

Figure 2 Chromosome structure in the proposed method

In the proposed chromosome structure, an n-member array is used to represent n features. The amount of each gene on this chromosome corresponds to one feature. The value of each gene is equal to 0 or 1. If the value of the

gene in question is 0, it means that no feature is selected, and if its value is equal to 1, it means that the feature is selected. The state space of the genetic algorithm can be developed with a series of 0s and 1s.

3.1.2 Fitness function and selection strategy

As mentioned in the genetic algorithm, the new population replaces the current population. To create a new population, a certain number of superior chromosomes are first selected by elitism and then included unchanged in the new population. The members of the new population are then generated based on the cross-over and mutation operators on the chromosomes of the current population. For this purpose, the parents are selected from the current population based on a selection strategy. The cross-over and mutation operators are then applied to the new population. In the genetic algorithm, the degree of competence of each chromosome is calculated based on the fitness function. The fitness of each chromosome has a great influence on its survival. In the selection phase, a pair of chromosomes is selected as the parent of the current generation to produce a child for the next generation. The operator thus selects an interface between two generations. The scoring function used to score the selected feature subset is the k-nearest neighbors algorithm (KNN).

3.2. ***Phase 2: Prediction using multilayer perceptron neural network***

After the redundancy features of the data are removed, the optimal data set is transmitted to the multilayer perceptron neural network. Neural networks are a class of machine learning algorithms that attempt to predict the process of converting inputdata into output data through regression. This means that they can predict the output when they receive new input data. The number of layers in multi-layer perceptron neural networks is limited and usually set by the user. In each layer, a set of neurons processes the data received from the previous layer and transmits its output to the next layer. In most applications, only one input layer is used and the number of neurons is set by the user. In addition to the input layer, there is an output layer whose number of neurons corresponds to the number of outputs of the entire network. A multilayer perceptron network uses sigmoid functions in the first layer and the same function in the output layer.

3.2.1 Data segmentation

A multilayer neural perceptron network divides the received input data into three parts:

1) Training data: This data is very specific and is used to train the neurons. This part of the data is fed to the neurons to regulate the sigmoid functions.
2) Validation data: This data, unlike the training data, is uncertain. This means that the output of this data is secret and only the input is given to the neurons. This data is used to evaluate the function of the neurons during the training process. In fact, the data determines the time of completion of the training process. During the training process, if the error rate of the validation data falls below the threshold or exceeds a certain threshold, the training process is terminated.
3) Test data: This data is uncertain, like the validation data. The difference between them and the validation data is that they are given to the neurons after the training process. This data determines the error rate of the prediction of the multilayer perceptron network.

3.2.2 Neural network training

The neural network training process begins after the redundant features of the data have been removed and used as input to the neural network, and by determining the partitioning of the data and the number of neurons. At this

point, the neural network tries to adjust the sigmoid function of each neuron to get the best estimate of the input-output data. The training of the neural network continues until one of the stopping conditions is reached. The stopping conditions are:

1) Number of specific iterations
2) Specific training time
3) The desired performance
4) Specific validation number
5) Specific gradient

If one of the above cases occurs, the training process is stopped and the trained neural network is terminated.

4. Results And Discussion

This section presents the results of the experiments applied to the proposed method. The results of the method presented in[25] are compared with the results of the proposed method. The following section presents the data set, the evaluation criteria, and the results obtained, followed by the analysis of these results.

4.1. *Data set*

The dataset used for the experiments is the KDD Cup, which includes 41 features and 4969 samples. The features of each session include duration, protocol type, service, active flags, number of bytes sent and received, session location and more. In addition to the 41 characteristics, there is a special characteristic that indicates the session class. The feature class represents a healthy session or an attack session. In fact, each session belongs to one of the two classes: "healthy" or "attacking.".

4.2. *Evaluation criteria*

After running each classification algorithm (including the neural network algorithm used in this paper), each sample is assigned to a class. Based on the correct classification, you can evaluate the functioning of the classifier. Each of the evaluation criteria attempts to show the distance between the proposed class and the correct class. As with many previous studies, we use common metrics such as precision, recognition, accuracy, and F_Measure to compare the results. Before presenting the evaluation criteria, we first give the computational requirements for these criteria. To calculate these criteria, we first determine the values of TP, TN, FP, and FN. Suppose we have a dataset. After classifying this dataset and comparing it with the optimal classification, one of the following four cases is proposed:

- If data a and b samples are in the same cluster in the proposed classification and in a cluster in the ideal classification, the TP increases by one.
- If data a and b samples are in the same cluster in the proposed classification but not in an ideal classification in a cluster, the FP increases by one.
- If data a and b samples are not in the same cluster in the proposed classification and are in an ideal classification in a cluster, FN increases by one.

If data a and b samples are not in the same cluster in the proposed classification and are not in the same cluster in the ideal classification, TN increases by one.

1) Precision: is a general criterion for measuring the performance of the proposed algorithm, which is calculated as follows.

$TP/(TP + FP)$

2) Recall is a general criterion for measuring the performance of the proposed method. The recall is calculated according to the following equation:

$TP/(TP + FN)$

3) Accuracy: As the name suggests, this criterion determines the accuracy of the proposed classification. Equation (3–4) is used to determine the accuracy of the proposed classification.

$(TP + TN)/(TP + TN + FP + FN)$

4) F_Measure: This criterion is determined on the basis of the Precision and Recall criteria. In the proposed method, we use the equation below to calculate F_Measure.:

$(2 * precision * recall)(precision * recall)$

4.3. *Evaluation results*

The evaluation results for two different scenarios are presented in the following sections. In the first scenario, the dataset is assessed with different parts, while in the second scenario, the training and test datasets are assessed with different sizes. The results for each of these scenarios are presented below.

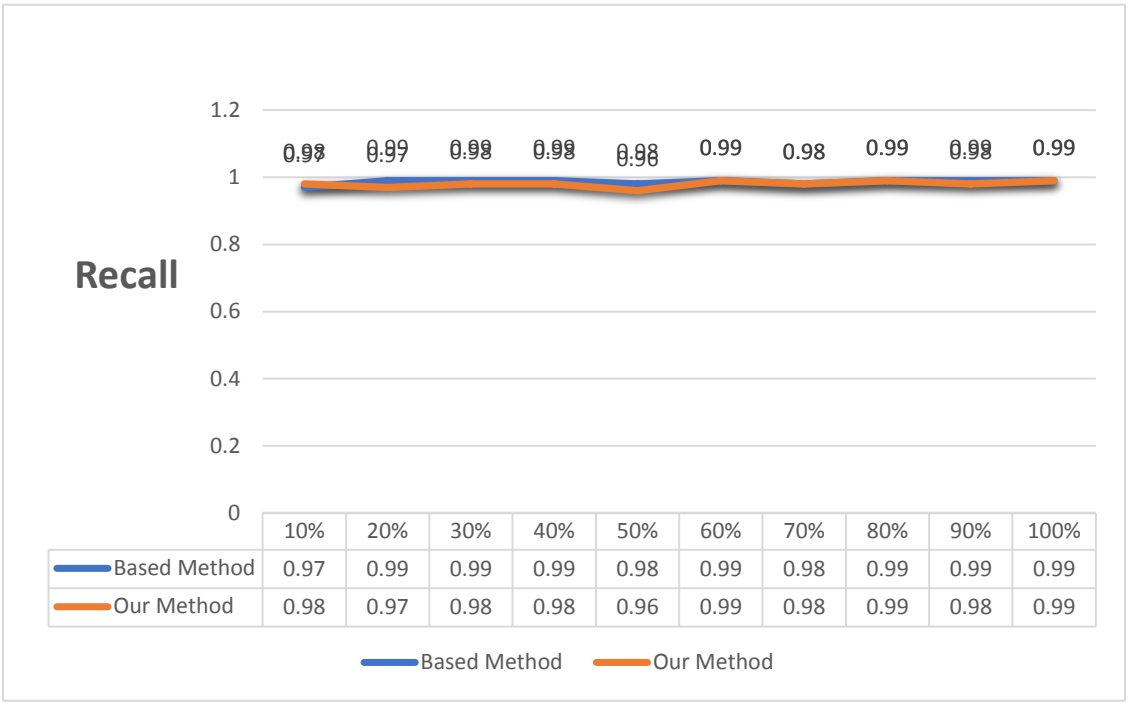
4.3.1 Scenario 1

As mentioned earlier, in the first scenario, the data set is divided into 10%, 20%, 30%, and so on. Each of the comparable methods is evaluated in these sections. In the results, this dataset to be evaluated is divided into 70% and 30% sections for training and testing. The results of these evaluations are shown in Figures 3 to 6.

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Based Method	0.95	0.66	0.65	0.68	0.99	0.67	0.98	0.68	0.67	0.68
Our Method	0.94	0.99	0.98	0.98	0.98	0.99	0.97	0.69	0.98	0.68

Figure 3 Comparison of precision criteria on data sets with different sizes

The results of pilot , shown in Figure 3, indicate that the proposed method produced better results. But in some cases, the results of the two methods were the same, so the superiority of the basic method over the proposed method is not obvious.



	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Based Method	0.97	0.99	0.99	0.99	0.98	0.99	0.98	0.99	0.99	0.99
Our Method	0.98	0.97	0.98	0.98	0.96	0.99	0.98	0.99	0.98	0.99

Figure 4 Comparison of Recall criteria on data sets with different sizes

The results show that the proposed method has obtained better results than the basic method in only 1 case, i.e, a data set with a size of 10%, but in other cases, the basic method has provided better results.

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Based Method	0.94	0.66	0.65	0.68	0.98	0.67	0.98	0.68	0.67	0.68
Our Method	0.94	0.98	0.98	0.97	0.96	0.99	0.97	0.69	0.97	0.68

Figure 5 Comparison of Accuracy criteria on data sets with different sizes

The results show that in all cases, the proposed method gave better results than the basic method. However, in some cases, the results of the two methods were the same, so that the superiority of the basic method over the proposed method is not obvious.

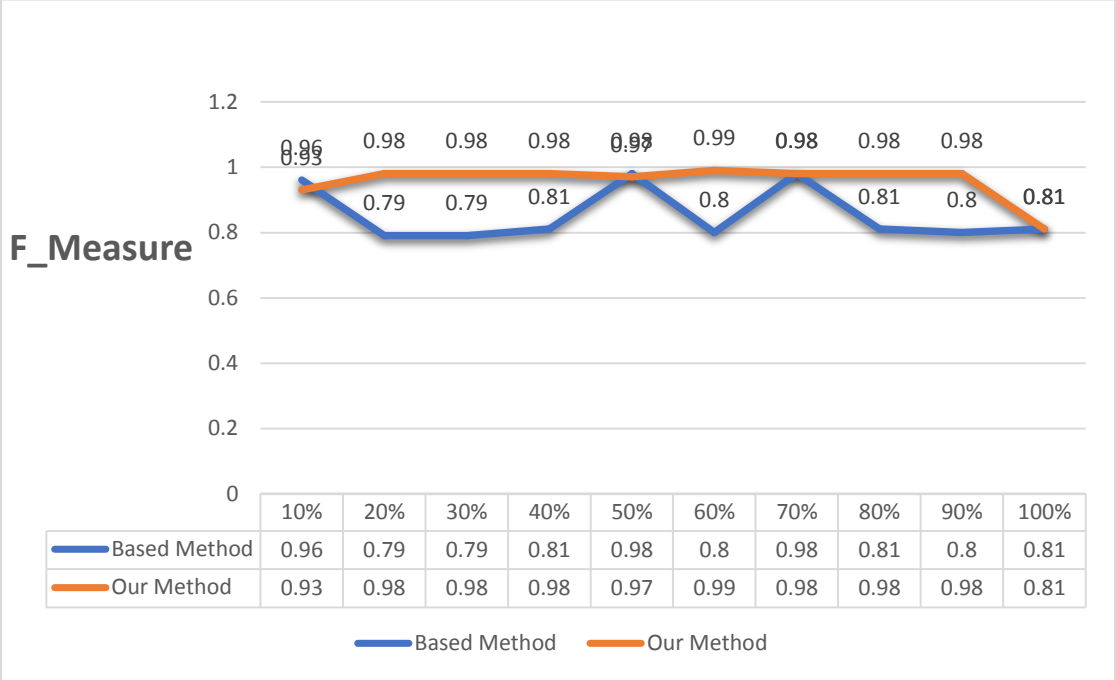


Figure 6 Comparison of F_Measure criteria on data sets with different sizes

As expected, the proposed method achieves better results than the basic method in every respect.

4.3.2 Scenario 2

In this scenario, 100% of the dataset is used, but the segmentation of the test set and the training set has been changed. In these results, the data set is divided into 30%, 40%, ... 80% sections for training. The results for this scenario are shown in Figures 7 to 10.

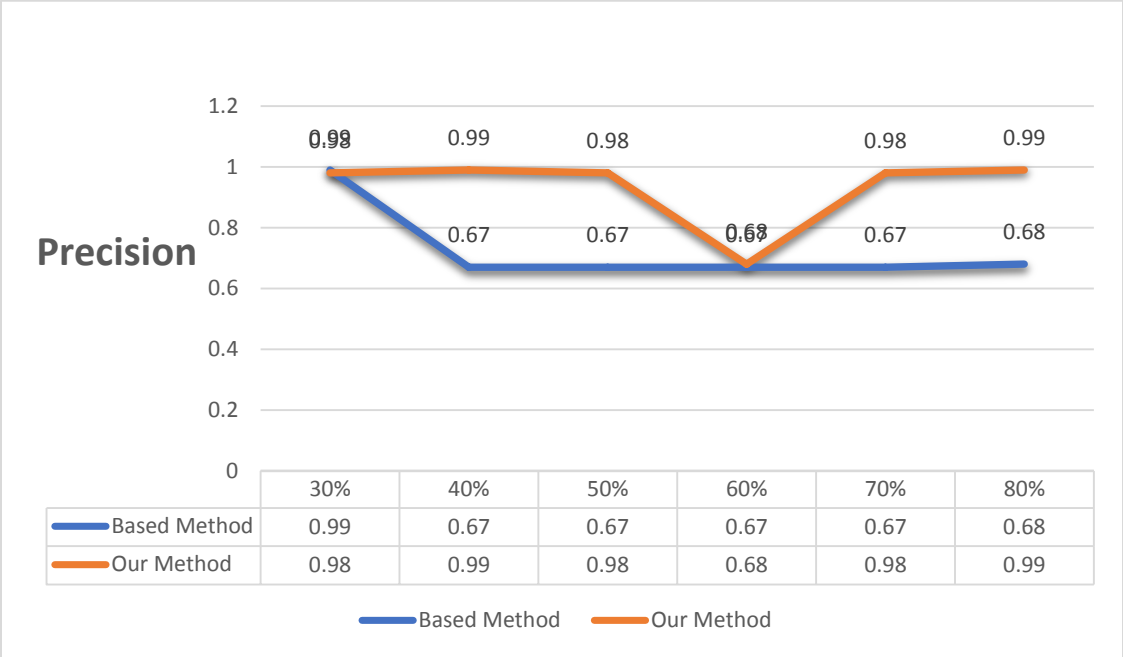


Figure 7 Comparison of precision criteria on data sets with different sizes

The precision obtained by the proposed method is always better than the basic method. In only two cases does the base method obtain the same values as the proposed method.

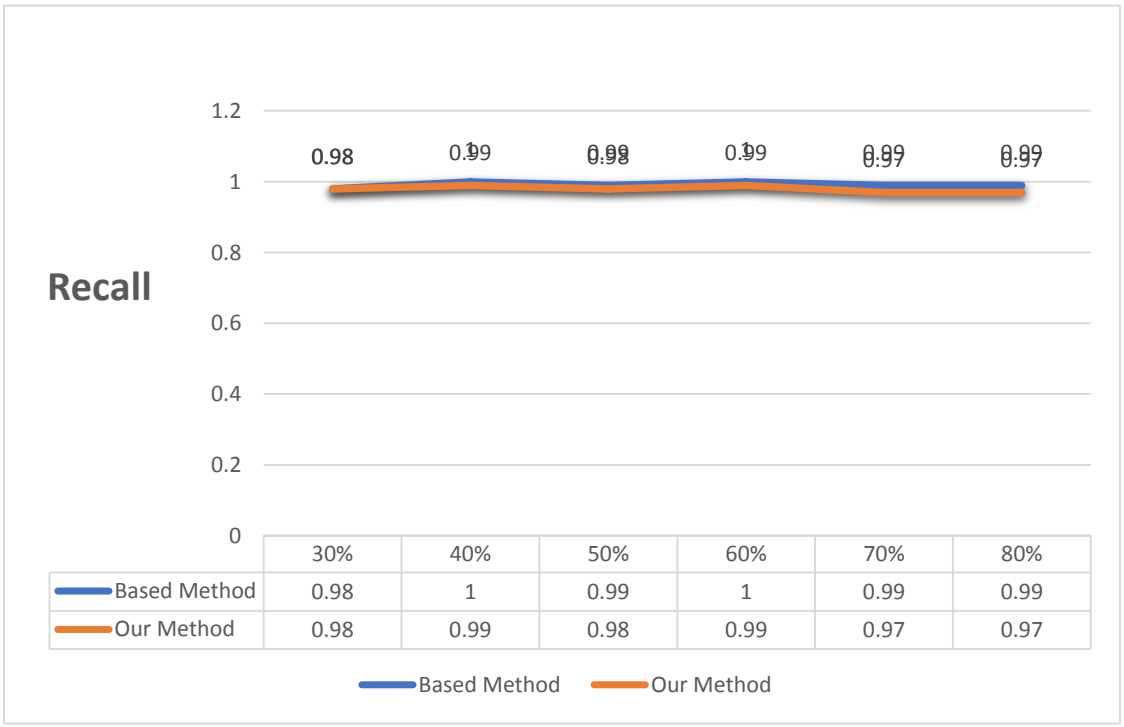


Figure 8 Comparison of Recall criteria on data sets with different sizes

The hit rate achieved with the proposed method is (with the exception of 30% of cases) always better than that of the basic method.

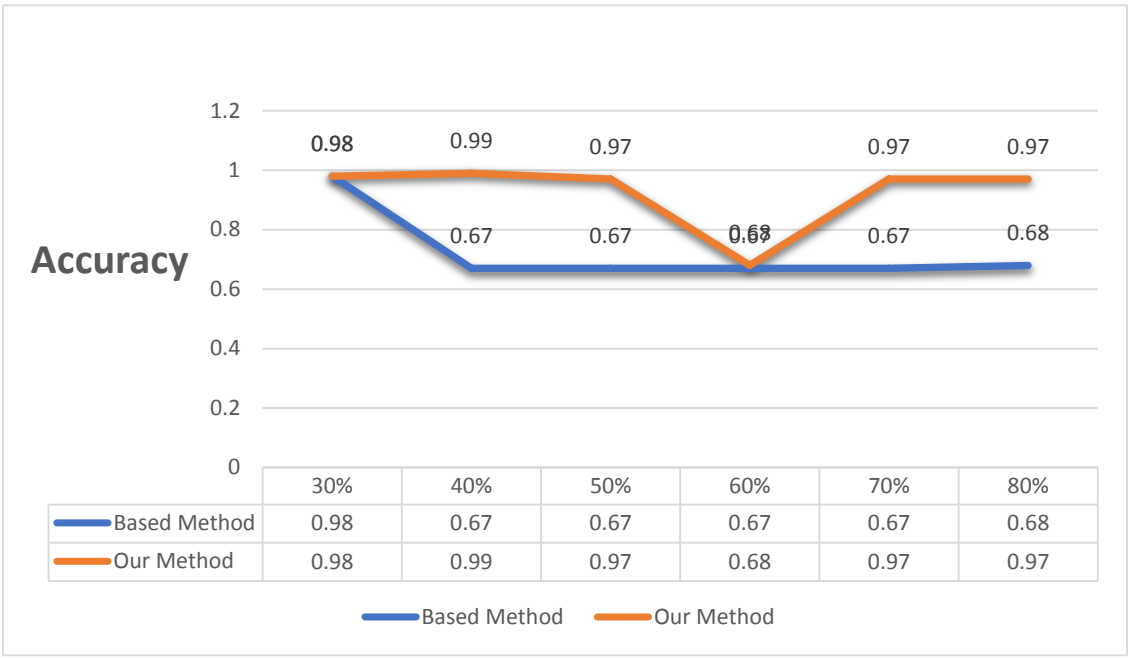


Figure 9 Comparison of Accuracy criteria on data sets with different sizes

The results in Figure 9 are similar to those in Figure 8, and the superiority of the proposed method is quite obvious.

Figure 10 Comparison of F_Measure criteria on data sets with different sizes

Considering that the F_Measure parameter is determined on the basis of precision and recall criteria and that the superiority of the proposed method on these two parameters has already been proven, it can be argued that the proposed method has also achieved better results on F_Measure. For all values of the quantity to be assessed, the proposed approach has better results than the basic method.

4.4. *Comprehensive comparison*

Tables 2 and 3 illustrate the results of the two scenarios, each based on the actual values in this section.

Table 2 — **Results of scenario 1**

Precision

Train Size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Based Method	0.95	0.66	0.65	0.68	0.99	0.67	0.98	0.68	0.67	0.68
Our Method	0.94	0.99	0.98	0.98	0.98	0.99	0.97	0.69	0.98	0.68

Recall

Train Size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Based Method	0.97	0.99	0.99	0.99	0.98	0.99	0.98	0.99	0.99	0.99
Our Method	0.98	0.97	0.98	0.98	0.96	0.99	0.98	0.99	0.98	0.99

Accuracy

Train Size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Based Method	0.94	0.66	0.65	0.68	0.98	0.67	0.98	0.68	0.67	0.68
Our Method	0.94	0.98	0.98	0.97	0.96	0.99	0.97	0.69	0.97	0.68

F_Measure

Train Size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Based Method	0.96	0.79	0.79	0.81	0.98	0.80	0.98	0.81	0.80	0.81
Our Method	0.93	0.98	0.98	0.98	0.97	0.99	0.98	0.98	0.98	0.81

Table 3 — **Results of scenario 2**

Precision

Train Size	30%	40%	50%	60%	70%	80%
Based Method	0.99	0.67	0.67	0.67	0.67	0.68
Our Method	0.98	0.99	0.98	0.68	0.98	0.99

Recall

Train Size	30%	40%	50%	60%	70%	80%

Based Method	0.98	1.00	0.99	1.00	0.99	0.99
Our Method	0.98	0.99	0.98	0.99	0.97	0.97

Accuracy

Train Size	30%	40%	50%	60%	70%	80%
Based Method	0.98	0.67	0.67	0.67	0.67	0.68
Our Method	0.98	0.99	0.97	0.68	0.97	0.97

F_Measure

Train Size	30%	40%	50%	60%	70%	80%
Based Method	0.98	0.80	0.80	0.80	0.80	0.81
Our Method	0.98	0.99	0.98	0.81	0.98	0.98

5. Conclusion

This paper presents an approach to detect DDoS attacks in IoT networks. The proposed approach consists of two basic phases. In the first phase, the session set is given a dimension reduction algorithm. Consequently, the data dimensions are reduced using a method based on a genetic algorithm with the k-nearest neighbor algorithm (as a fitness function) as the fitness function. After the first phase, the data is transferred to a multilayer perceptron neural network. The predictability of this classification distinguishes destructive from healthy sessions. The evaluation is based on the criteria of precision, recall, accuracy, and F_Measure in two different scenarios. In these scenarios, the size of the training dataset varies between 10% and 100% and the size of the test dataset between 30% and 80%. The results of the evaluations are compared with the approach presented in reference[25]. The results of the proposed method and the basic method are compared using the parameters of specificity, sensitivity, and accuracy. The results show that the proposed method performs better than the base method on the criteria of precision, accuracy, and F_Measure. The baseline method only performed better on the recall criterion.

References

1. Nuiaa, R.R., et al., *A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks.* International Journal of Electrical and Computer Engineering, 2022. **12**(2): p. 1869.
2. Alsammak, I.L.H., et al., *A model for blockchain-based privacy-preserving for big data users on the internet of thing.* Indonesian Journal of Electrical Engineering and Computer Science, 2022. **26**(2): p. 974-988.
3. Khempetch, T. and P. Wuttidittachotti, *DDoS attack detection using deep learning.* IAES International Journal of Artificial Intelligence, 2021. **10**(2): p. 382.
4. Valdovinos, I.A., et al., *Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions.* Journal of Network and Computer Applications, 2021. **187**: p. 103093.
5. Singh, N. and V. Jotwani, *A Hybrid Intrusion Detection System for Botnet attack with Data Technique.* NeuroQuantology, 2022. **20**(8): p. 4839-4847.
6. Tripathi, N. and N. Hubballi, *Application layer denial-of-service attacks and defense mechanisms: a survey.* ACM Computing Surveys (CSUR), 2021. **54**(4): p. 1-33.
7. Lysenko, S., et al., *Detection of the botnets' low-rate DDoS attacks based on self-similarity.* International Journal of Electrical and Computer Engineering, 2020. **10**(4): p. 3651.
8. Mohammed, A.J., M.H. Arif, and A.A. Ali, *A multilayer perceptron artificial neural network approach for improving the accuracy of intrusion detection systems.* IAES International Journal of Artificial Intelligence, 2020. **9**(4): p. 609.
9. Abomhara, M. and G.M. Køien. *Security and privacy in the Internet of Things: Current status and open issues*. in *2014 international*

conference on privacy and security in mobile systems (PRISMS). 2014. IEEE.

10. Snehi, M. and A. Bhandari, *Vulnerability retrospection of security solutions for software-defined Cyber–Physical System against DDoS and IoT-DDoS attacks.* Computer Science Review, 2021. **40**: p. 100371.
11. Zhang, Y., et al., *A survey on neural network interpretability.* IEEE Transactions on Emerging Topics in Computational Intelligence, 2021.
12. Roy, S., et al. *A survey of game theory as applied to network security*. in *2010 43rd Hawaii International Conference on System Sciences*. 2010. IEEE.
13. Li, M., I. Koutsopoulos, and R. Poovendran. *Optimal jamming attacks and network defense policies in wireless sensor networks*. in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. 2007. IEEE.
14. Beckery, S., et al. *Applying game theory to analyze attacks and defenses in virtual coordinate systems*. in *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*. 2011. IEEE.
15. Buttyan, L. and J.-P. Hubaux, *Security and cooperation in wireless networks: thwarting malicious and selfish behavior in the age of ubiquitous computing*. 2007: Cambridge University Press.
16. Alpcan, T. *A decision and game theoretic approach to networked system security with applications to power grid*. in *Proceedings of the Eleventh Australasian Information Security Conference-Volume 138*. 2013.
17. Tambe, M., et al. *Game theory for security: Key algorithmic principles, deployed systems, lessons learned*. in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2012. IEEE.
18. Sanyal, S., A. Shelat, and A. Gupta. *New frontiers of network security: The threat within*. in *2010 Second Vaagdevi International Conference on Information Technology for Real World Problems*. 2010. IEEE.
19. Chen, L. and J. Leneutre, *A game theoretical framework on intrusion detection in heterogeneous networks.* IEEE Transactions on Information Forensics and Security, 2009. **4**(2): p. 165-178.
20. Alpcan, T. and T. Basar. *An intrusion detection game with limited observations*. in *12th Int. Symp. on Dynamic Games and Applications, Sophia Antipolis, France*. 2006.
21. He, W., et al. *A game theoretical attack-defense model oriented to network security risk assessment*. in *2008 International Conference on Computer Science and Software Engineering*. 2008. IEEE.
22. Alpcan, T. and T. Basar. *A game theoretic analysis of intrusion detection in access control systems*. in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*. 2004. IEEE.
23. Fallah, M., *A puzzle-based defense strategy against flooding attacks using game theory.* IEEE transactions on dependable and secure computing, 2008. **7**(1): p. 5-19.
24. Chen, Z., *Modeling and defending against internet worm attacks*. 2007, Georgia Institute of Technology.
25. Doshi, R., N. Apthorpe, and N. Feamster. *Machine learning ddos detection for consumer internet of things devices*. in *2018 IEEE Security and Privacy Workshops (SPW)*. 2018. IEEE.