Designing a participatory modelby using ensembleintelligence algorithms to detect security anomalies in wireless sensor-based IoT networks

Fatemehkhezri, Department of Computer Engineering, south Tehran Branch, Islamic Azad University, Tehran, Iran.

Behzad Habili, Department of Computer Engineering, Iranian university, Tehran, Iran.

Abstract

Background: With the development and expansion of technologies and computing technologies, today the frontiers of knowledge are developing rapidly. One of the most important factors in starting these technologies is the formation of a computer network.

purpose: Network traffic is very large and this leads to high data size and increased noise and makes it very difficult to extract meaningful information to detect abnormal events. Network training to detect abnormalities helps to identify the time of the attack. Early detection of attacks improves the stability of a system.

Methods: The goal of this study is to design a proposed model to identify and detect malware attacks with appropriate accuracy and reduce the rate of malfunctions. To meet these goals, we used participatory machine learning algorithms.

Results: According to the results, our proposed model has a better performance in intrusion detection of wireless sensor network attacks than the compared algorithms of decision tree, random tree and Naïve Bayes.

Conclusion: Due to recent advances in data mining, the use of participatory model in identifying and detecting attacks of wireless sensor networks is very effective.

Keywords: ParticipatoryImprovedLearning, EnsembleLearning, Identification and Detection of Malicious Attacks

1.Introduction

Due to the expansion of computer networks in the world, the issue of network security is one of the topics that has attracted the attention of researchers [1]. Detection of user intrusion or malware is one of the important factors for establishing security within computer networks which controlling network traffic and analyzing the behavior of users within the networkare the main purposes of these systems. To research and improve performance in this field, many researches have been done during these years and various methods and techniques have been used to identify intrusions into the network. Preventive security measures such as firewalls, user authentication, and encryption are not enough to secure networks such as the Internet of Things. Therefore, the use of intrusion detection systems methods and technologies can be a very good defense mechanism for important and vulnerable points of the network. Extracting patterns to detect the misbehavior of network users and predict their behavior is difficult and time consuming due to the large amount of data available in network traffic. Due to the different characteristics of the data in IoT network traffic, there may be data that has incorrect connections to each other, which in turn prevents the detection of true intrusion. For this reason, examining data and their properties and selecting the appropriate features can increase the speed of systems and increase their accuracy [2].

The Internet of Things is a new paradigm that integrates the Internet and physical objects, objects that belong to different domains such as home automation, industrial processes, human health monitoring and environmental monitoring [3] - [5]. The Internet of Things deepens the presence of Internet-connected devices in our daily activities and brings many benefits, as well as security-related challenges [3]-[5]. For more than two decades, intrusion detection systems (IDS) have been an important tool for protecting IoT networks and information systems. However, conventional IoT methods are difficult to use in IoT because of certain features, such as resource-constraineddevices, specific protocol stacks, and standards. Intrusion detection systems report any suspicious activity by monitoring the traffic passing through the network. Just like a guard in an organization who reports suspicious behavior by observing the movement of people and activities so that the organization can prevent that destructive behavior if necessary.

The order of presentation of the rest of the contents of this article is as follows. Previous studiesare discussed in Section 2. Section 3 examines the proposed

protocol. Section 4 presents the findings of the article. In Section 5, we present the open issues and results of this article.

2. Previous Studies(Literature Review)

Intrusion Detection Systems (IDSs) can be categorized as Network-Based IDS (NIDS) and Host-Based IDS (HIDS). Network-based IDS (NIDS) connects to one or more network segments and monitors network traffic to detect malicious activity. Host-based IDS (HIDS) connects to a computer device and monitors malicious activity taking place inside the system. Unlike NIDS, HIDS not only analyze network traffic, but also analyze system calls, process execute, file-system changes, inter-process communications, and application reports [6].

IDS approaches may also be categorized as signature-based, anomaly-based, or feature-based.

Signature-based diagnostics are consistent with the network's current behavior against predefined attack patterns. In the IoT, signatures are first assigned and then stored on the device, and each signature corresponds to a specific attack. It is usually easy to use the signature-based methods that require a signature for each attack. The continuation of this section is a review of selected articles in this field [7].

Amin et al. [8] have demonstrated a dynamic coding mechanism for implementing a signature-based distributed IDS in IP-USN (IP-based sensor network). In this method, signatures of different lengths are first sent to their relative bloom filters. The output of Bloom filters is a small array of signature codes. After sending the contents of the received packets to the Bloom filters, the pattern is checked and if there is a match, the sensor node stops processing the packet and sends the alarm signal with the signature code to the well to confirm the match of the Bloom filter[9]. The Snort signature set is used to evaluate the performance of the proposed method.

Oh et al. [10] have proposed a security mechanism that uses a pattern matching engine for malicious nodes in the Internet of Things. On devices with resource-constrained, limitations on computing power and memory reduce performance. So two strategies such as assisted relocation and early decision making are defined in the article. These methods practically reduce matching operations in the IoT environment. In patterns with equal prefix values, character matching is performed and the information obtained is used to detect early termination of matching operations [11]. The results show that this method improves the

performance, especially when the sample values are very large. A Raspberry Pi and Omnivision 5647 sensor are used to evaluate the proposed IDS for IoT devices.

Sun et al. [12] have proposed a cloud-based framework for intrusion detection called CloudEyes. On the client side, CloudEyes implements a lightweight scanning agent that uses a combination of signature components to abruptly reduce the range of exact matching.The CloudEyes framework uses a lightweight monitoring operator and a combination of signature components to reduce the precision matching range. The cloud server implements the database and provides a summary of the signatures in a reversible layout.

3.Proposed Protocol

In this section, we will describe and present the proposed method according to the flowchart that is presented, and in the following, the steps that are in the flowchart will be fully explained with details.

In general, our main goal in this article is to detect malware attacks on networks. The following section presents the steps related to the proposed method.

Malware datasets are first pre-processed for use in RapidMiner or MATLAB software. Due to the large amount of information we may need to remove duplicate samples in the data set. Eliminating duplicate samples not only does not have a negative effect on the created model, but also reduces the learning time of the model and reduces duplicate calculations.Then, using the k-th method of evaluation, the data is divided into k parts to create training and test data. In this method, the data is divided into k blocks. In each recurrence, one k block is considered as test data and the rest as training data. Training and test data are entered into the ensemble learning system or all of the algorithms used in this system one by one. In the following, we will describe and present the proposed method according to the flowchart that is presented.

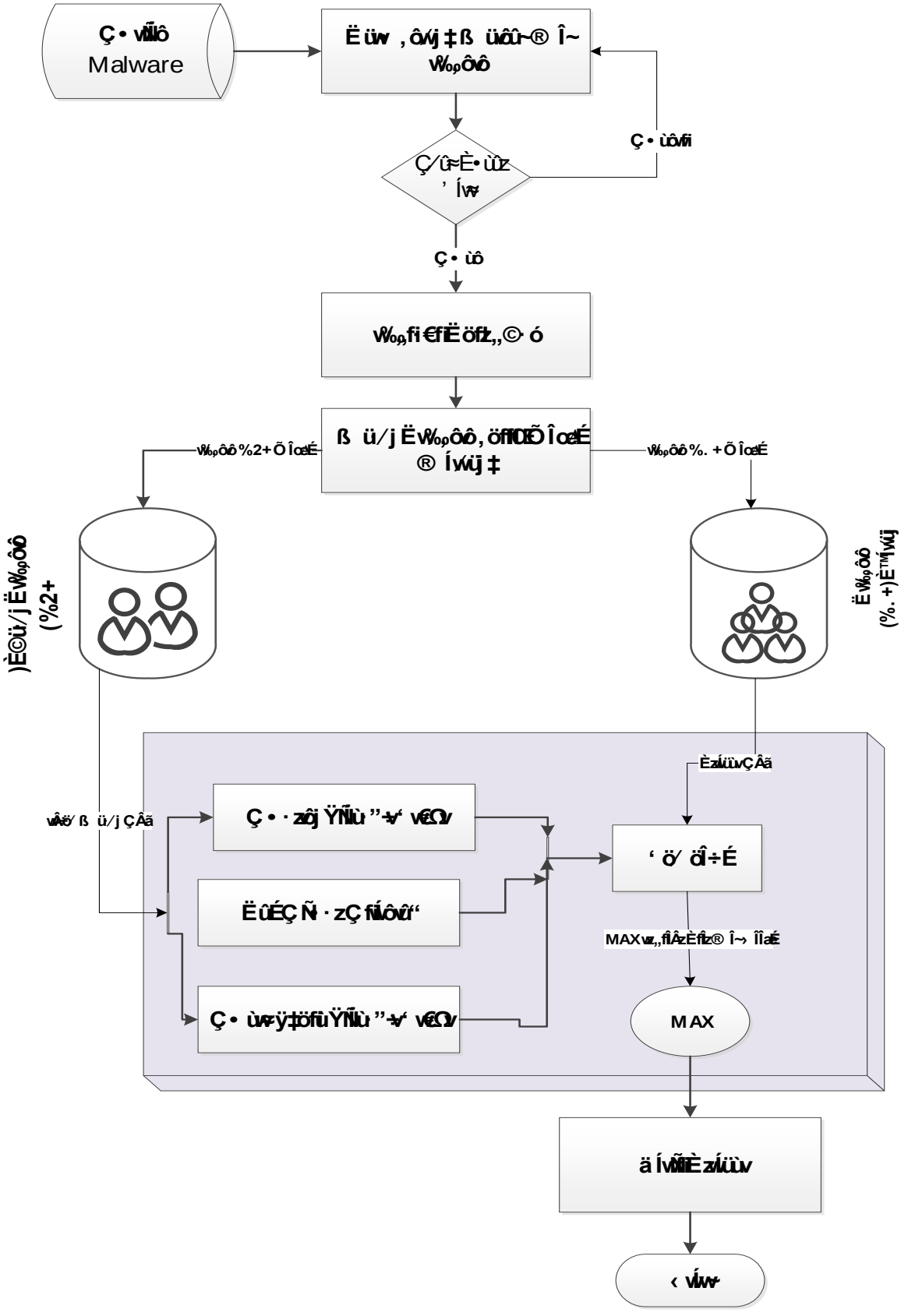


Figure1: Proposed Method Flowchart

In this study, a combination of Adaboost, Random Forest and Gradient Boosted Trees techniques is used to detect malware attacks in the network by removing unbalanced specimens. Two of the most important algorithms used to detect malware attacks are Adaboost and Random Forest. The procedure for combining and boosting these two algorithms is that first the relevant data is analyzed by each algorithm and the relevant predictions are adopted. Finally, using one of the maximum strategies, the best candidate is selected and returned as the final answer. The most important goal of data boosting is to combine the results of the two methods and extract the best answer in each step. In the following, a complete description of the proposed algorithm is presented.

After separating the test data and the training data, it is necessary to perform a preprocessing on the data and also to convert the data format to acceptable formats of the tools used in next step. So after downloading the Malware dataset, we first copy the relevant data to a text file. Then copy all this data again and after generating an Excel file, open the corresponding Excel file and paste all the data by clicking on a cell.

3-1-Separation of test and training data

In order to produce a model based on trainingdata, it is necessary to separate part of the data as trainingdata and part of the data as test data. Therefore, about 70% of the data is used as trainingdata and 30% of the data as test data. Training data are used to teach the boosting algorithm and model production, and test samples are used to evaluate the proposed method and compare it with other methods.

3-3-7-Applying boosting algorithm on data sources

So the data is convertedto an acceptable format for data mining software. In this step, the Voting algorithm is used so that the algorithms of Adaboost, Random Forest and Gradient Boosted Trees are applied in combination to the input of the problem which is part of the research innovations. It is applied to training data and then we enter test data into this model. This algorithm uses the voting system to build the model, so that the majority of votes between the algorithms always go to the output. For example, Adaboost and Random Forestmay considera recordas an attack, but Gradient Boosted Trees man notconsider it as an attack, but based on the majority of votes, the record is considered as an attack. The flowchart below shows the format and structure of the Voting systemwhich is a combination of Adaboost, Random Forest and Gradient Boosted Trees.

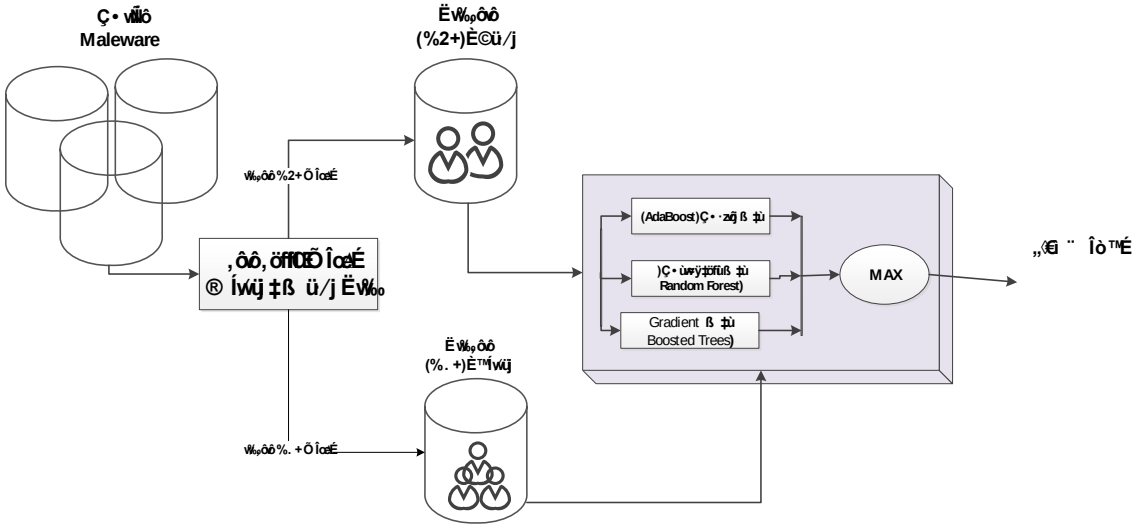


Figure2: Proposed SystemStructure

As can be seen in the figure above, first the main dataset is entered into the system and data are divided by a data separator intotwo categories to producepredictive models and to evaluate the generated model. Training data which constitute 70% of data, are used to generate the model and the test data which constitute 30% of the data, are used to evaluate the accuracy and error of the models generated by the proposed algorithm. While the method of separating data into training and test samples is described in the previous section, this section does not re-explain it. Therefore, after entering the training data, learning models are taught under the supervision of Adaboost, Random Forest and Gradient Boosted Trees. By teaching the mentioned algorithms, the relevant models are taughtto evaluate the new data according to the trainingthey have received andto provide the relevant predictions.

Therefore, each model in the boosting system has a specific output that indicates the type of prediction. As shown in the figure, the output of each of the random forest and adaboost models is entered into the maximum logic, and this strategy selects the best response from the outputs of each of the algorithms and sends it to the output. By applying such a combined system, we will see desirable results.

3-2- Evaluation criteria

In this paper, several criteria have been used to evaluate the proposed method with other methods, which are: [13]:

Equation 1-3 $precision = \frac{TP}{TP+FP}$

The above equation is used to check the accuracy of the proposed method. The TP parameter indicates the number of abnomalies that have been correctly detected. And the FP parameter also indicates the number of samples that are not anomaly but the proposed models consider that sample as an anomaly [14].

Equation 2-3 $ReCall = \frac{TP}{TP+FN}$

The above equation indicates the recall rate of the proposed method. The FN parameter indicates the number of samples that are not anomaly and are correctly detected.

Equation 3-3 $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

The above equation is used to calculate the accuracy, the only parameter that is not described is TN, which indicates the number of samples that have been anomaly but they are detectedas non- anomaly. Finally, the error rate of the proposed method is calculated based on the following formula.

Equation 4-3 $Error = 100 - \frac{TP+TN}{TP+TN+FP+FN}$

The proposed method in this research has been implemented using RapidMiner version 9 simulator. The following table also shows the specifications of the system which implemented proposed method andevaluated its results.

Table 1: System specifications for simulation and evaluation of results

Specifications	Hardware / Software
Windows 11	**operating system**
64-bit operating system	**Operating system type**
24 GB	**RAM**
Intel Processor - Number of Cores 7 (Core ™) i5 CPU)	**Processor**
Rapid Miner data mining software	**modeling tool**
UNSW-NB15 data set	**Data source used**

In this article, the UNSW dataset will be used. This dataset is used to detect malware attacks on the network. This dataset consists of two parts: training data and test data. Training data is used to generate the model and test data is used to examine and evaluate the proposed model.

This dataset has nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

In the proposed method, using the Voting method,RadomForest, GradientBoostedTrees and AdaBoostalgorithms are combined and a new model isgeneratedso that in this method, voting is done in allthreealgorithms and according to the outputs of each algorithm, the maximum number of votes isdetermined as the final result.

The Vote operator is a nested operator, meaning it has a subprocess. This subprocess must have at least two learners. This operator creates a classification model or regression model depending uponExampleSet and learners. This operator uses a majority vote (for classification) or the average (for regression) on top of the base learner predictions provided in itssubprocess.

In the classification operation, all available operators in the Vote operator subprocess accept the given ExampleSet and generate a classification model. To predict an unknown example, the Vote operator applies all classification models from its subprocess and assigns the predicted class with maximumvotes to the unknown example. Similarly, in case of regression task, all the operators in the Vote operator subprocess accept the given ExampleSet and generate a

regression model. To predict an unknown example, the Vote operator applies all the regression models from its subprocess and assigns the average of all predicted values to the unknown example.

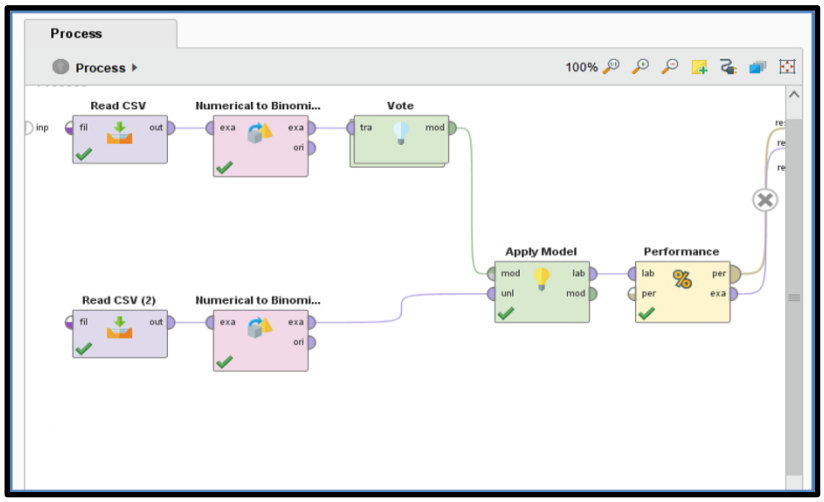


Figure 3: Simulation of the proposed method of the first step

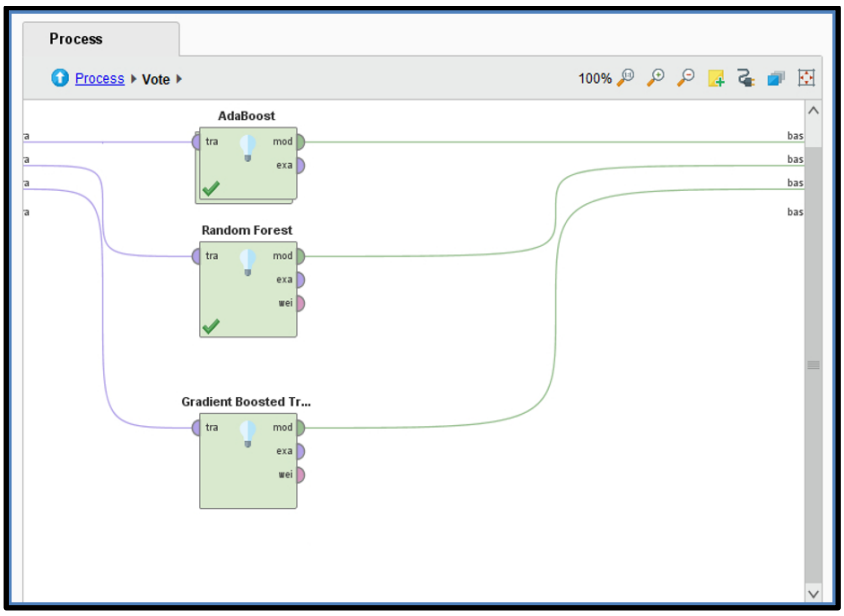
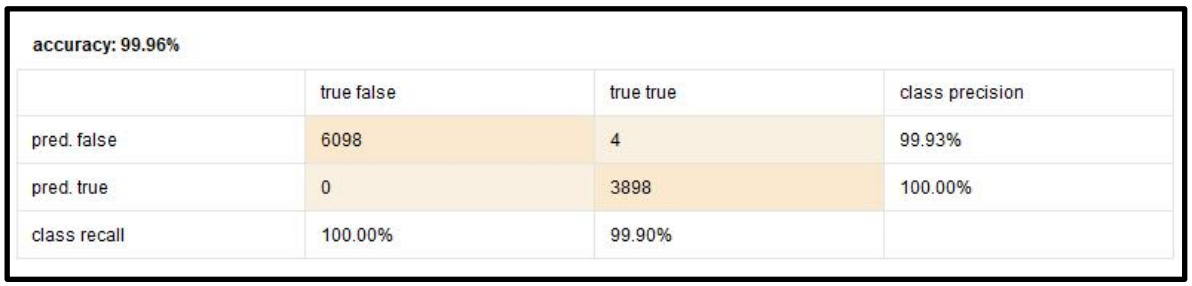


Figure 4: Simulation of the proposed method of the second step inside the Vote operator itself.

The accuracy of the results of this proposed algorithm is shown in the figure below.

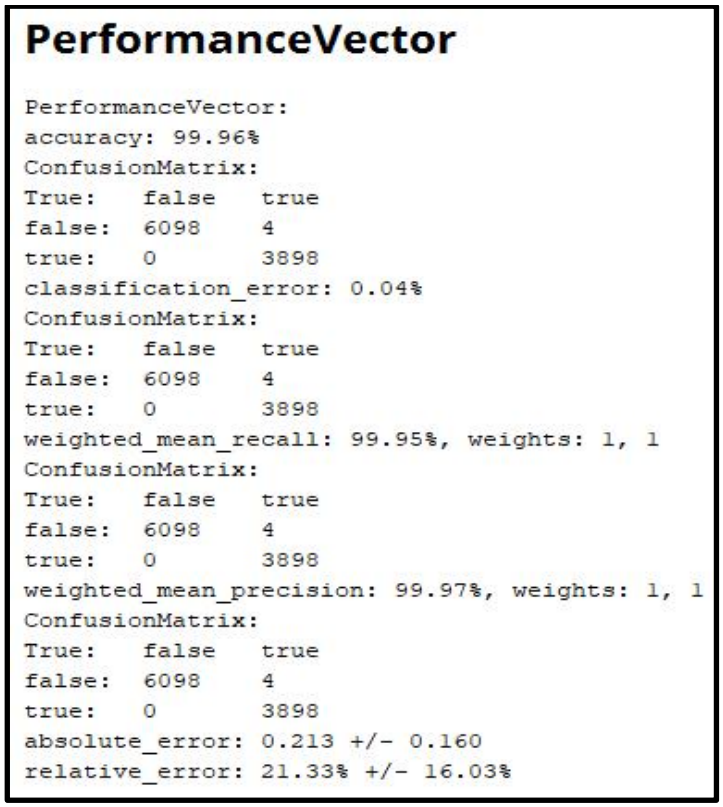


accuracy: 99.96%

	true false	true true	class precision
pred. false	6098	4	99.93%
pred. true	0	3898	100.00%
class recall	100.00%	99.90%	

Figure 5: Accuracy obtained from simulation of the proposed method.

According to the figure above, the accuracy of the proposed algorithm is 99.96%. The following figure shows additional information about the accuracy of the algorithm and its error.



PerformanceVector

```
PerformanceVector:
accuracy: 99.96%
ConfusionMatrix:
True:   false   true
false:  6098    4
true:   0       3898
classification_error: 0.04%
ConfusionMatrix:
True:   false   true
false:  6098    4
true:   0       3898
weighted_mean_recall: 99.95%, weights: 1, 1
ConfusionMatrix:
True:   false   true
false:  6098    4
true:   0       3898
weighted_mean_precision: 99.97%, weights: 1, 1
ConfusionMatrix:
True:   false   true
false:  6098    4
true:   0       3898
absolute_error: 0.213 +/- 0.160
relative_error: 21.33% +/- 16.03%
```

Figure 6: Results obtained from the simulation of the proposed method.

As it can be seen, the accuracy of the proposed method and the modeling error are shown, respectively. The accuracy of the proposed method is 99.96% and its error rate is 0.04%, the accuracy rate is 99.97%, and the recall rate is 99.95%.

4-Findings

The table below shows a comparison of the accuracy of the proposed method compared to other methods. Therefore, in this section, the results obtained in the proposed method are comparedwith these methods.

Table 2: The Comparison of the results of the proposed method with the presented methods

Gradient Boosted Trees	Random Forest	AdaBoost	proposed method	
60.98	72.85	87.24	99.96	**Accuracy**
39.02	27.15	12.76	0.04	**Error**
30.49	79.48	87.68	99.97	**Accuracy rate**
50.00	77.74	89.54	99.95	**Recall rate**

As it can be seen, the proposed method is more optimal than other methods in terms of accuracy, error, accuracy rate and recall rate. The following figure shows a comparison of the accuracy of the proposed method compared to other methods proposed so far.

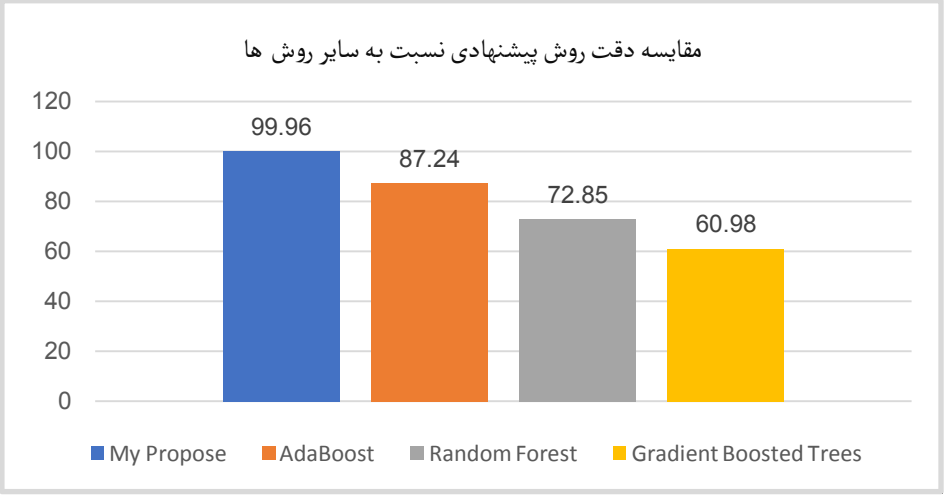


Figure 7: Comparing the accuracy of the proposed method with other methods

As it can be seen, the accuracy of the proposed method for detecting attacks in cloud computing networks is 99.96%, which is a very acceptable accuracy and is more optimalthan other methods that have been proposed so far. The following figure shows the error comparison of the proposed method compared to other methods that have been proposed so far.

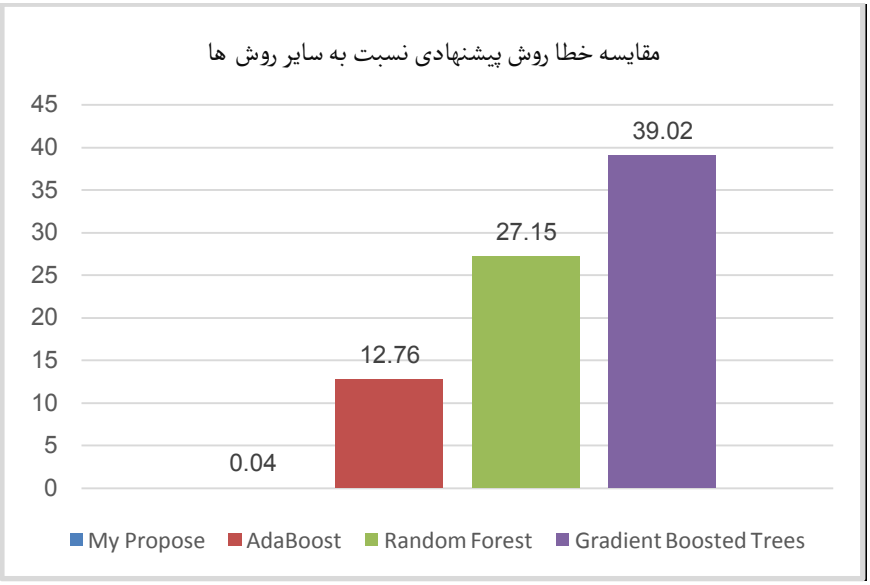


Figure 8: Comparison of the error of the proposed method with other methods

As it can be seen in the proposed method, the amount of attack detection error in cloud computing networks is 4%, which is very small and can not disrupt the development of these systems.

Comparing the results of the proposed method, it is found that the proposed method has improved by about 38.98% compared to the Gradient Boosted Tree method in the NUSW dataset. It is also found that the proposed method has improved by about 27.11% compared to the random Forest method in the NUSW dataset. On the other hand, the proposed method has improved by about 12.72% compared to the Adaboost method in the NUSW dataset.

5-Discussion and conclusion

The simulation of the proposed method starts with the preprocessing of Malware data and removes the junk and unused values of this data, which will increase the accuracy and speed in this system, and as a result, it will be more accurate to detect attacks. Finally, the system results are stored in an Excel file.

The proposed method is based on Adaboost, Random Forest and Gradient Boosted Tree (GBT) algorithms.

In this section, we will mention some suggestions that can be used as new designs and ideas to improve the performance of the proposed method in this article. Some of these suggestions are:

- Using a combination of support vector machine algorithms and neural network algorithms in order to detect malware attacks in the network and compare the results with current research findings

- Using feature selection algorithms such as particle swarm algorithms, etc. in order to select outstanding features and finally to detect malware attacks on the network

- Using optimization algorithms such as genetic algorithms, bee, SSO and PSO algorithms and other similar algorithms to improve the results and the identified detections.

References

[1] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21954–21961, 2017.

[2] A. M. Vartouni, S. S. Kashi, and M. Teshnehlab, "An anomaly detection method to detect web attacks using stacked auto-encoder," in *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 2018, pp. 131–134.

[3] M. Sedighimanesh, H. Z. Hesami, and A. Sedighimanesh, "Routing algorithm based on clustering for increasing the lifetime of sensor networks by using meta-heuristic bee algorithms," *Int. J. Sensors Wirel. Commun. Control*, vol. 10, no. 1, pp. 25–36, 2020.

[4] K. M. sedighimanesh mohammad, Zandhessami Hessam, Alborzi Mahmood, "Energy Efficient Routing-Based Clustering Protocol Using Computational Intelligence Algorithms in Sensor-Based IoT," *J. Inf. Syst. Telecommun.*, vol. 33, no. 9, 2021, doi: 10.52547/jist.9.33.55.

[5] K. M. Sedighimanesh Ali, Zandhessami Hessam, Alborzi Mahmood, "Optimal Clustering-based Routing Protocol Using Self-Adaptive Multi-Objective TLBO For Wireless Sensor Network," *J. Inf. Syst.*

Telecommun., vol. 34, 2021, doi: 10.52547/jist.9.34.113.

[6] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 78, pp. 267–287, 2017.

[7] N. Dey, A. E. Hassanien, C. Bhatt, A. Ashour, and S. C. Satapathy, *Internet of things and big data analytics toward next-generation intelligence*, vol. 35. Springer, 2018.

[8] Y. Li, H. Chen, M. Lv, and Y. Li, "Event-based k-nearest neighbors query processing over distributed sensory data using fuzzy sets," *Soft Comput.*, 2019, doi: 10.1007/s00500-017-2821-2.

[9] W. Meng, W. Li, Y. Xiang, and K.-K. R. Choo, "A bayesian inference-based detection mechanism to defend medical smartphone networks against insider attacks," *J. Netw. Comput. Appl.*, vol. 78, pp. 162–169, 2017.

[10] P. Braca, R. Goldhahn, G. Ferri, and K. D. Lepage, "Distributed Information Fusion in Multistatic Sensor Networks for Underwater Surveillance," *IEEE Sens. J.*, 2016, doi: 10.1109/JSEN.2015.2431818.

[11] G. Elhayatmy, N. Dey, and A. S. Ashour, "Internet of Things based wireless body area network in healthcare," in *Internet of things and big data analytics toward next-generation intelligence*, Springer, 2018, pp. 3–20.

[12] H. Sun, X. Wang, R. Buyya, and J. Su, "CloudEyes: Cloud-based malware detection with reversible sketch for resource-constrained internet of things (IoT) devices," in *Software - Practice and Experience*, 2017, doi: 10.1002/spe.2420.

[13] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.

[14] S. Gurung, M. K. Ghose, and A. Subedi, "Deep learning approach on network intrusion detection system using NSL-KDD dataset," *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 3, pp. 8–14, 2019.