Proceedings of IAM, V.11, N.2, 2022, pp.71-86

USING OF DECISION TREE CLASSIFICATIONS METHOD FOR INTRUSION DETECTION IN INTERNET NETWORKS

M. SALATI[1], I.N. ASKERZADE[1], G.E. BOSTANCI[1], M.S. GÜZEL[1]

[1]Department of Computer Engineering, Ankara University, Turkey
e-mail: imasker@eng.ankara.edu.tr

Abstract: This study proposes a method for intrusion detection in the Internet networks through decision tree classification tools in machine learning. Furthermore, the grey wolf optimization (GWO) algorithm was adopted for feature dimensionality reduction and feature selection. The random forest, classification tree, and regression tree were employed to detect attacks on the Internet. The proposed method was then evaluated through the NSL-KDD dataset. According to the results, the random forest was more efficient than classification and regression trees.

Keywords: Internet, machine learning, attack detection, feature selection, grey wolf optimization (GWO), random forest, classification and regression trees.

AMS Subject Classification: 68T05, 68T20.

1. Introduction.

The Internet is now used extensively in many areas. Given the high effectiveness of the Internet in today's society, the Internet-based cyberspace faces various challenges, one of which is the detection of intrusion and attacks in such networks. Intrusion into the Internet networks can jeopardize privacy, urban security, and security of institutions. Hence, it is essential to detect attacks on the Internet. A method of detecting such attacks is to use machine learning tools. In recent decades, many security problems have arisen on the Internet and in computer systems due to the explosion in the use of networks. According to the Computer Emergency Readiness Team (CERT), intrusion into systems is unbelievably increasing on a yearly basis. Any kinds of destructive intrusion or attack can harm computer networks, systems, and information, probably resulting in serious events such as the violation of computer security policies such as confidentiality, integrity, and accessibility [11]. Threats to networks and information security are still among the noteworthy research areas, and there is rich literature on the analysis and classification of intrusion detection methods [1,3,15].
Intrusion detection systems try to detect anomalies by analyzing the information on activities in systems and networks. Emerging as events in systems, attacks can pose different levels of threats. These events emerge as the following forms:

1) Network packets
2) OS recalls
3) OS-conducted audits
4) Software status packets

Intrusion detection systems aim to analyze one or several sets of events and identify intruders. When an attack is detected, a warning is generated to inform the system official. The system events are analyzed in two ways to find intrusions:

1) Searching for anomalies [5]
2) Searching for misuses [15]

In search for anomalies, the data of all system activities such as the behaviors of users and programs are employed to generate a file containing information on the display of jobs in the system. After that, the intrusion detection system starts to identify the pattern of malicious activities. The Internet includes important subfields, among which the Internet of things (IoT) is a concept that has presented a noteworthy aspect of the Internet.

The IoT is a newly emerging technology used in different areas such as healthcare, transportation, and smart networks. The applications of the IoT are often diverse in many fields, for it can be used in tiny devices that can be placed on the skin or connected to home appliances [8]. The processing power and energy supply of the IoT devices are limited due to their subtlety and tininess as well as the fact that they should be mobile and often dependent on light batteries. With the growing demand and ever-increasing developments in the automated network and IoT systems, the IoT models are becoming more and more complicated on a daily basis [18-19]. The IoT is considered the Third Industrial Revolution [21]. In fact, the IoT is defined as the connection of computing devices embedded in everyday devices and data transfer via the Internet [10]. The IoT sensors/devices often collect and process spatiotemporal information on specific events and the environment to deal with different challenges [7-9]. The IoT has made objects smarter. Healthcare has become more intelligent, and communications have become more instructive. Hence, the IoT is used in nearly all areas: home applications, education, entertainment, energy distribution, financial affairs, healthcare, smart cities, tourism, and transportation [21].

A method of intrusion detection on the Internet (and its subfields) is to use machine learning tools, which have been used by many researchers. The next section addresses these applications. This paper also proposes a machine learning method based on feature selection and decision tree classifications for intrusion detection on the Internet. Section 2 reviews the research background, whereas Section 3 describes the proposed method and the applied tools. Finally, Section 4 reports the results.

2. Research Background.

This section reviews some of the relevant studies from recent years.

Given the increasing attacks on the Internet, developing an intrusion detection system has become a necessity for the security of systems. In most of the proposed intrusion detection systems, a database is employed to store the patterns of attacks. This database is also used for system protection. Gang Wang *et al.* [4] proposed a method for the automated detection of patterns stored and used in the data sources of an intrusion detection system.

Vijay Anand *et al.* (2020) [2] proposed a method based on the whale optimization algorithm (WOA) and genetic operators for intrusion detection in the wireless mesh networks. Based on feature selection, their proposed method was aimed at improving the WOA through genetic operators to prevent early binding. They proposed a wrapper-based method for feature selection and used a support vector machine classifier for data classification.

In Ref [21] proposed a method through the machine hybrid teaching learning based optimization-extreme learning for intrusion detection in the Internet networks. They pursued two goals: 1) analyzing the existing hybrid solutions and their constraints and 2) proposing a new solution called TLBO–ELM based on the firefly optimization algorithm and the fast learning networks.

With the increasing use of the IoT infrastructure in all areas, threats and attacks have been growing proportionally. These attacks and anomalies include the denial of service (DOS), data type probing, malicious control, malicious performance, scanning, spying, and misconfigurations that can cause failure in the IoT system. Accordingly, in Ref. [6] proposed a method based on machine learning (ML) algorithms to detect attacks and anomalies in the IoT. They employed classification tools such as logistic regression (LR), support vector machines (SVMs), and artificial neural networks (ANNs) and then evaluated the results in terms of accuracy, precision, and F-score on the dataset introduced in [20]. According to their results, machine learning methods can be efficient in anomaly detection.

Liu *et al.* (2018) [14] proposed a tracker for the on-off attack of a malicious network node in the industrial IoT site. Analyzing the on-off attack, they wanted to show that the IoT network could be attacked by a malicious node in the active-inactive state. Moreover, the IoT network behaves normally when its malicious node is in the inactive or off state. This system was developed for anomaly detection by using an optical probing routing mechanism and estimating the reliability of every neighboring node.

The intrusion detection system (IDS) is gaining in popularity through the use of machine learning methods, for it benefits from the advantage of self-updating in order to protect the network against any new kinds of attacks. In fact, the IoT is a newly emerging technology responsible for developing an automated system by connecting devices without human intervention. In the IoT-based systems, the wireless connections of multiple devices via the Internet will result in vulnerability to different security threats. Kumar *et al.* (2019) [11] proposed a method called the unified intrusion detection systems (UIDS) for IoT environments by adopting machine learning and classification tools. Their proposed method included data preprocessing (selecting samples, determining intrusive and normal samples, and selecting features). They also used decision tree classifiers such as CART, CHAID, C5, and QUEST. After that, they evaluated the proposed method on the UNSW-NB15 dataset [20]. Precision was the most important evaluation criteria in their work. According to the results, C5 proved to be the most efficient classifier with a precision rate of 89.76%.

Liu *et al.* (2018) [12] proposed a method based on the fuzzy logic to improve intrusion detection in the IoT by using the suppressed fuzzy clustering (SFC) algorithm and the principal component analysis (PCA) technique. For this purpose, they first classified data as high-risk and low-risk categories identified as high frequency and low frequency, respectively. At the same time, the detection frequency self-regulation was performed through the SFC algorithm and the PCA technique. Finally, the key factors affecting the algorithm were analyzed more deeply in a simulation. The results indicated that their proposed method was more compatible than the conventional method.

3. The Proposed Method.

This section describes the proposed method. For this purpose, the necessary tools of implementation are first presented, and the proposed procedure is then discussed.

3.1. Feature Selection.

In this paper, the grey wolf optimization (GWO) algorithm was used for feature selection based on the wrapper technique. In fact, the GWO algorithm is inspired by the behaviors of grey wolves in nature. Known as skilled hunters, grey wolves are at the top of the food chain. They often prefer to live in packs of five to 12 wolves [16]. The problem-solving agents are called wolves in the GWO algorithm; they move toward a prey (*i.e.*, an optimal solution) in packs.

After the initial population of wolves is generated in the GWO algorithm, the fitness values of wolves are calculated. The alpha, beta, and delta wolves that have the best fitness values are selected as leaders in the pack of wolves. The leader wolves determine the new positions of other wolves referred to as the omega wolves. After the positions of wolves are updated, their fitness values are calculated. If the wolves are transferred to better positions, then the new alpha, beta, and delta wolves are selected. Otherwise, no changes occur. This process is iterated until the algorithm meets the termination condition. The GWO algorithm consists of the following steps:

1- Initialization: The initial population of a wolf pack is generated to solve the problem.
2- If the termination conditions are met, go to Step 3; otherwise, repeat the following steps:
2-1- Calculate the fitness values of al wolves in the pack.
2-2- Determine the alpha, beta, and delta wolves.
2-3- Update the positions of all wolves in the pack in accordance with alpha, beta, and delta wolves.
2-4- Go to Step 2.
3- Select the alpha wolf as the final solution to the GWO.
4- End.

The objective function was calculated during the feature selection process in the following way. This study aimed to select features (delete reiterative and useless

features) and reduce the classification rate. Hence, Equation 1 was employed to determine the efficiency of the model generated by agents:

$$\text{Performance(Model)} = W_1 * Error(\text{Model}) + W_2 * \frac{SF_i}{TF} \quad (1)$$

Where *Error* denotes the random forest classifier's error, and SF_i refers to the number of the selected features (SF) in the *i*th agent. Furthermore, TF indicates the number of total features (TF) in the dataset. The smaller the value of $\frac{SF_i}{TF}$, the better the agent. Finally, $W_1 = 0.8$ and $W_2 = 0.2$.

3.2. Decision Tree.

The classification and regression tree (CART) was used in this study. In fact, a decision tree is a classification algorithm in which the samples are classified in a way that the tree moves downward from the root and reaches the nodes in the end. Every internal or non-leaf node is characterized by a feature, which raises a question regarding the input sample. Based on the existing range, a decision is made upon what procedure should be selected for the next step or what branch should be taken next. In the decision tree, the leaf nodes are the final classes. For example, a sample moves from the root toward a leaf to determine which class the sample belongs to. Finally, it ends up in a leaf node (class) [10].

This method was called the decision tree because this process indicates the decision-making process of determining the class of an input sample. The decision trees can describe the relationships of a dataset in a way that is perceivable for humans. They can also be used for classification and prediction tasks.

This decision-making structure can also be introduced as mathematical and arithmetic techniques that help describe, classify, and generalize a dataset. The data are given in records like (x, y) = (x1, x2, x3…, xk, y). The independent variables (x1, x2, …, xk) are employed to perceive or classify the dependent variable (Y).

In a decision tree, different types of attributes are divided into classified attributes and real attributes. The classified attributes are the ones that accept two or more discrete values (*i.e.*, symbolic attributes), whereas the real attributes receive their values from the real numbers.

Decision trees are used mainly to achieve the following goals in classification:

- The input data should be classified as correctly as possible.
- Developing a model through the training data, they should properly predict new data classes.
- If new training data are added, the decision tree should be easily developed (*i.e.*, it should be expandable).
- The resultant tree structure should be as simple as possible.

The following steps are necessary for designing a decision tree:

- Selecting the right decision tree.
- Selecting the features of interest for decision-making in each of the middle nodes.

- Selecting the decision-making rule or strategy used in each of the middle nodes.

Selecting different rules or strategies can lead to the generation of different trees.

3.3. Random Forest.

Random forests were introduced by Leo Breiman (see [15]) inspired by the results reported by Amit and Geman. In fact, random forests can be employed to determine the class of the target variable. This process is known as "classification" but is also called "regression" if it is used to predict a continuous target vector. Similarly, the predicting variables or features can be of the nominal or numerical types.

Random forests are attractive from an arithmetic perspective because:

- They naturally classify both regression and classification.
- They are relatively fast in training and prediction.
- They depend only on one or two regulation parameters.
- They estimate the generalization error.
- They can be used directly to solve high-dimensional problems.
- They can easily be implemented in a parallel framework.

Random forests are also attractive from a statistical perspective because:

- They measure the importance of a feature.
- They weight different classes.
- They control and manage the lost values.
- They use visualization.
- They detect outliers.

Random forests operate by combining classifiers. In fact, the baseline classifiers in a random forest are decision trees. The RF-based model operates by averaging the outputs of all baseline decision trees. The random forest generates many decision trees. For any new sample, the outputs of each tree used in the random forest are calculated. The final result is then determined for the input sample by voting on the outputs.

3.4. The Proposed Method.

This section presents the proposed procedure for intrusion detection in the Internet networks. Figure 1 demonstrates the general procedure. Accordingly, the appropriate data were identified by reviewing the previous studies. The data used in this study had been employed in many of the previous studies. These data are addressed in the next section. After data collection, since some data have no numerical values, they were converted into acceptable forms through Excel for use in classes. The data were then divided into training and test classes.

The training data were employed to develop classification models, whereas the test data were used to calculate the results. Finally, the efficiency criteria were calculated.

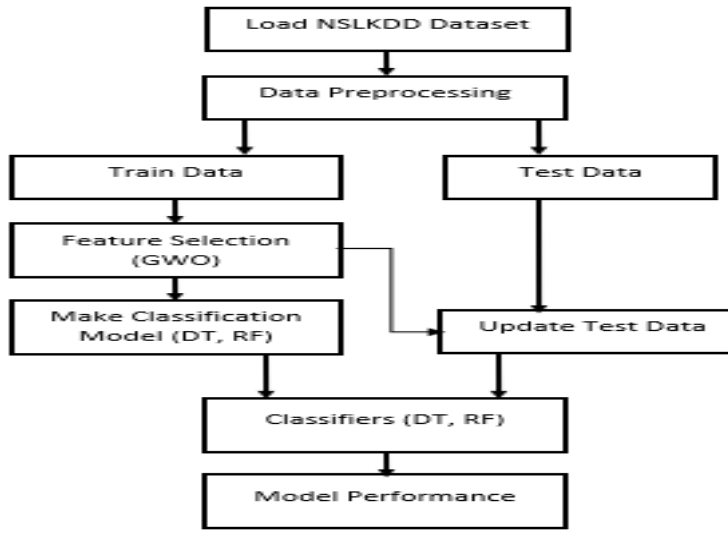

Figure 1. The proposed method.

4. Results.

This section reports the results of the proposed method. For this purpose, the designated dataset is first introduced. After that, the evaluation criteria are presented. Eventually, the results of the proposed method are explained.

4.1. Dataset.

The NSL-KDD dataset was used in this study. A modified version of the KDD CUP99 dataset, NSL-KDD has 41 features and includes 22 attacks, which are divided mainly into denial of service (DOS), user to root (U2R), remote to local (R2L), and probe groups. Hence, this dataset has five classes, four of which include attacks, whereas the fifth one includes healthy data. In this paper, two classes of healthy data and attack data were first considered. Therefore, the classification is of the two-class type in this step. The efficiency of each attack was then analyzed.

4.2. Evaluation Criteria.

The classification efficiency evaluation criteria were used in this paper to analyze efficiency. Accordingly, a confusion matrix was employed to calculate the evaluation criteria. Table 1 demonstrates the confusion matrix.

Table 1. The confusion matrix for the efficiency analysis of classification algorithms

		Real class of a sample	
	Total samples	Positive	Negative
Outputs of classifiers (estimated class)	Positive	TP	FP
	Negative	FN	TN

According to the confusion matrix, the following criteria are employed to calculate the efficiency of a proposed model:

$$Accuracy = \frac{TP+TN}{N} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F_{score} = \frac{2*TP}{2*TP+FP+FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

These criteria are employed to evaluate classification algorithms. The greater the values of these criteria, the more efficient an algorithm. The next subsection reports the results of the proposed method.

4.3. Outputs.

The GWO algorithm was used for feature selection. The GWO algorithms were initialized as below:
The population of wolves: 12, The maximum number of iterations: 50
Figure 2 demonstrates the convergence diagram of the GWO algorithm.

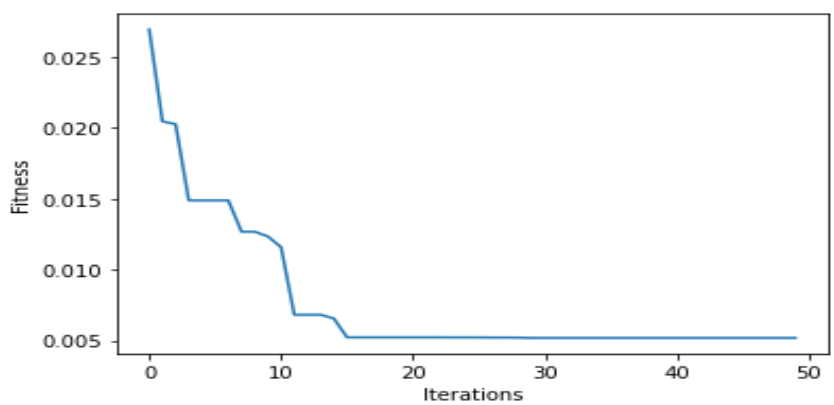

Figure 2. The divergence diagram of the GWO algorithm for feature selection.

According to Figure 1, the data were divided into training (70%) and test (30%) segments in this study. This method of division is known as the holdout technique. This subsection reports the outputs of the proposed method on training and test data. The results are also reported before and after feature selection.

Table 2. reports the results of the proposed method on the training data. The following points should be taken into account:

- The number of baseline learners in the random forest was considered 100.
- The number of the selected features was considered 9.

Table 2. The results of the proposed method on the training data

	Before feature selection		After feature selection	
	CART	RF	CART	RF
Precision	99.99	100	99.97	100
Recall	99.97	100	99.94	100
Accuracy	100	100	100	100
F-score	99.99	100	99.97	100

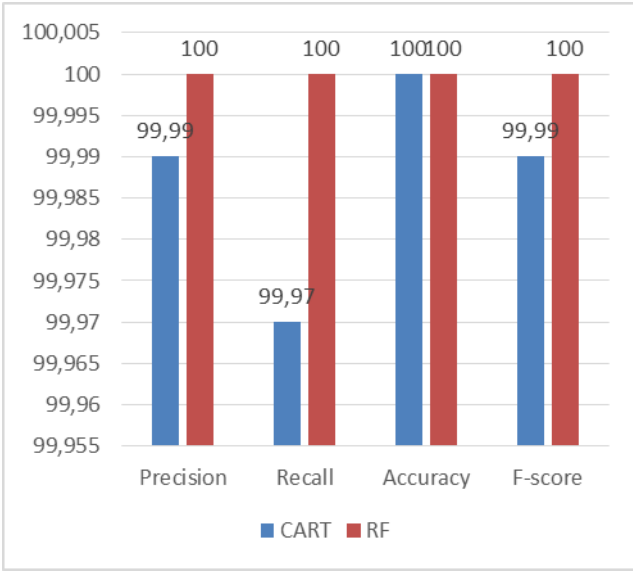

Figure 3. Before feature selection

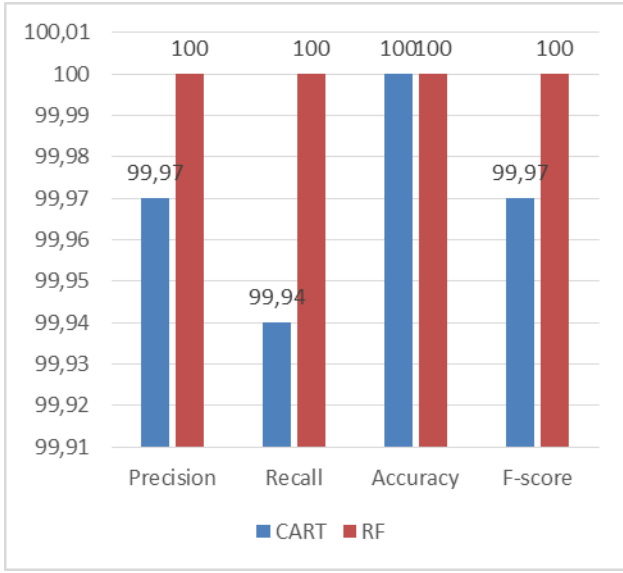

Figure 4. After feature selection

The results were reported in percentage in Table 2 and Table 3. The greater these values (and closer to 100), the more efficient the model. According to Table 2, the following conclusions can be made:

- The random forest was more efficient than the CART.
- Apparently, classifiers were acceptably efficient.
- The efficiency of the CART was better before feature selection; however, since the number of features declined by nearly 77% in feature selection, the complexity of the CART decreased after feature selection.
- These results (on the training data) cannot be employed to analyze the efficiency of the proposed method. In fact, they cannot be cited. Therefore, the results on the test data should be analyzed.

Table 3. reports the results on the test data:

Table 3. The results of the proposed method on the test data

	Before feature selection		After feature selection	
	CART	RF	CART	RF
Precision	99.83	99.99	99.80	99.93
Recall	99.80	100	99.87	100
Accuracy	99.87	99.97	99.73	99.87
F-score	99.83	99.98	99.80	99.93

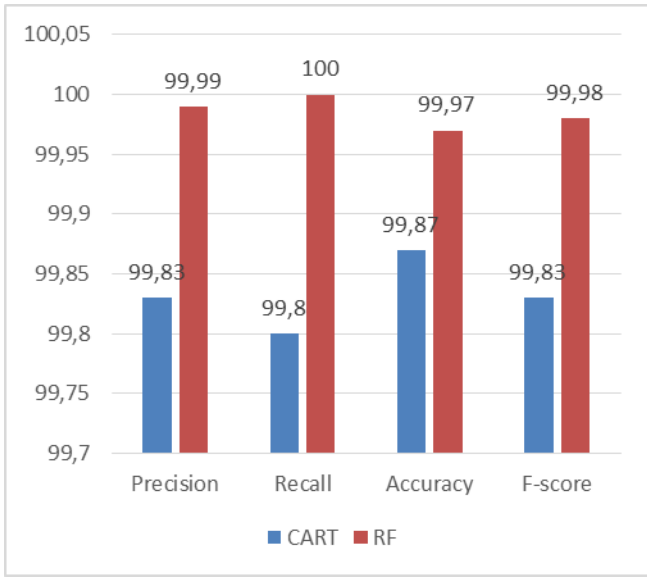

Figure 5. Before feature selection

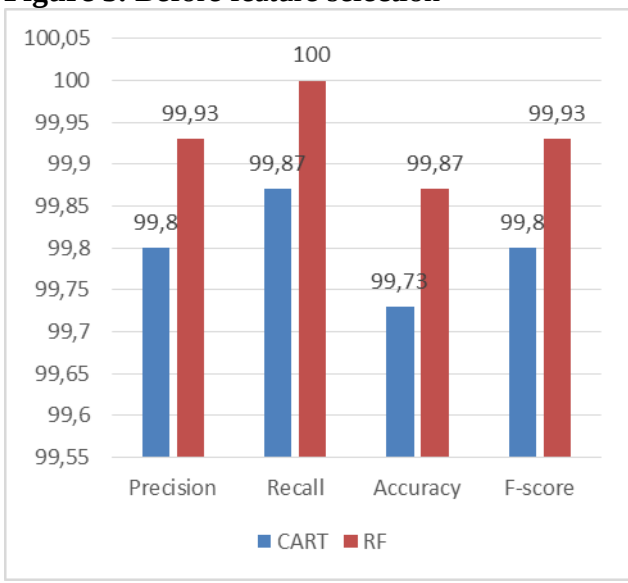

Figure 6. After feature selection

The results of the multiclass mode are reported as below for each class. In this case, it should be noted that there were five classes.

Table 4. The results of the proposed method on the test data in the multiclass mode

Attack Class	Normal		Dos		Probe		R2L		U2R	
Algorithm / Metric	RF	DT	RF	DT	RF	DT	RF	DT	RF	DT
Accuracy	98.62	98.56	99.76	99.76	99.76	99.69	99.1	99.07	99.81	99.77
Precision	97.76	97.61	99.84	99.84	99.53	99.17	97.11	96.76	95.04	88.62
F1-Score	98.37	98.31	99.64	99.64	98.94	98.65	96.01	95.90	94.26	92.77
Recall	98.98	99.01	99.43	99.43	98.36	98.13	94.94	95.05	93.5	97.32

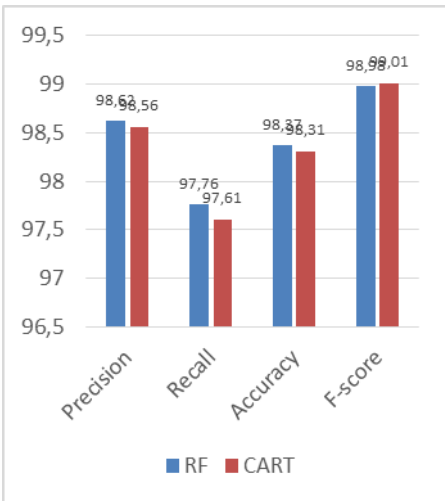

Figure 7. Normal

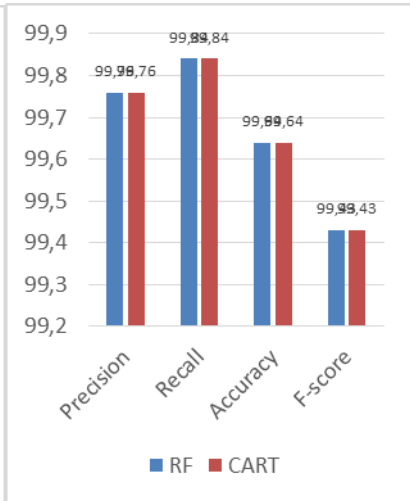

Figure 8. Dos

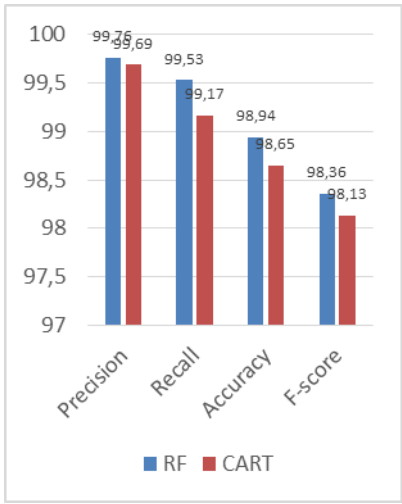

Figure 9. Probe

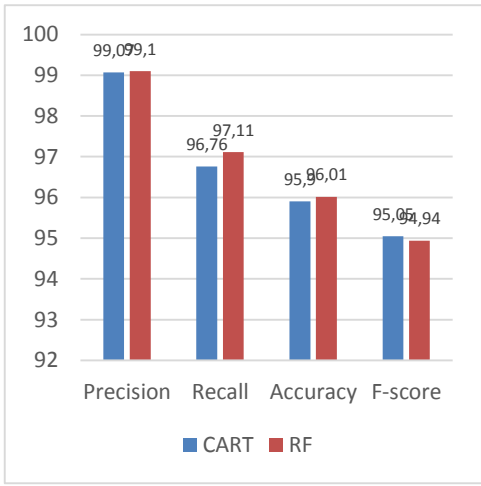

Figure 10. R2L

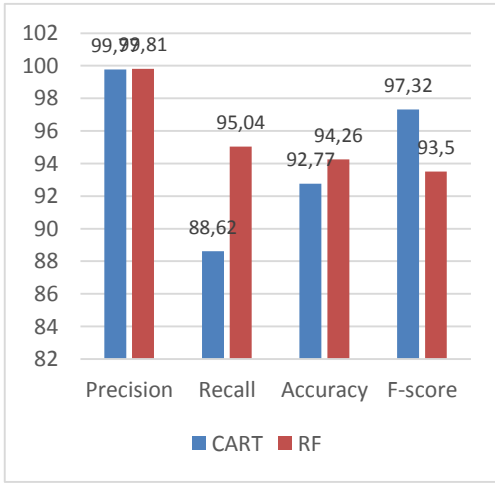

Figure 11. U2R

According to the results on the test data, the proposed method was acceptably efficient in intrusion detection. The following conclusions can be made:

- The machine learning algorithms can be efficient in intrusion detection if they are used correctly.
- The random forest algorithm is an ensemble learning method. It is more efficient than the single CART in intrusion detection.
- Although feature selection affects the efficiency of models, it reduces their complexity.
- According to the analysis of results on training and test data, the proposed model had no overfitting.

The next section draws a conclusion on the proposed method.

5. Conclusion.

Since the IoT is growing in our lives on a daily basis, it has resulted in certain challenges, a major one of which is the problem of security and intrusion in particular. Intrusion detection is of great importance in the IoT. This study proposed a method for intrusion detection in the IoT based on the machine learning algorithms. For this purpose, feature selection and classifiers were employed. The results indicated the high efficiency of these algorithms in intrusion detection. However, the following research avenues can be considered for further analysis:

- Using more recent and more comprehensive datasets
- Employing hybrid classification methods and integrating different classifiers
- Selecting features through metaheuristic algorithms
- Optimizing random forest parameters (*i.e.*, the number of trees) to acquire better results

References

1. Amer, Suhair H , Jr John A Hamilton, Input Data Processing Techniques in Intrusion Detection Systems Short Review, GJCST, Vol.9, No.18, (2009), pp.5-18.
2. Anand, Vijay & Devaraj, D., A Novel Feature Selection Method Using Whale Optimization Algorithm and Genetic Operators for Intrusion Detection System in Wireless Mesh Network, IEEE Access, Vol.8, (2020), pp.56847-56854. 10.1109/ACCESS.2020.2978035
3. Bosman G. Iacca, Tejada A., Wörtche HJ., Liotta A., Spatial anomaly detection in sensor networks using neighborhood information, Information Fusion, Vol.33, (2017), pp.41–56.
4. Gang W., et al., A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, Expert Systems with Applications, Vol.37, No.9, (2016), pp.6225-6232.
5. Ghosh, Anup K., James Wanken, Frank Charron, Detecting anomalous and unknown intrusions against programs, Computer Security Applications Conference, Proceedings. 14th Annual. IEEE, (1998).
6. Giovanni Vigna, Richard A. Kemmerer, NetSTAT: A network-based intrusion detection system, Journal of Computer Security, Vol.7, No.1, (1999), pp.37-71.
7. Gubbi J., Buyya R., Marusic S., Palaniswami M., Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems, Vol.29, No.7, (2013), pp.1645–1660.
8. Hasan M., Islam M.M., Zarif I., Hashem M.M.A., Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches, Internet of Things, Vol.9, (2019), pp.34-43.
9. Intel, U.N. IDC, A Guide to the Internet of Things Infographic, (2015).
10. Kohavi R., The power of decision tables, Machine Learning: Proceedings of the Eighth European Conference on Machine Learning ECML95, Lecture Notes in Artificial Intelligence, Springer Verlag, 914, Berlin, Heidelberg, NY, (1995), pp.174–189.
11. Kumar V., Das A.K., Sinha D., UIDS: a unifed intrusion detection system for IoT environment, Evolutionary Intelligence, (2019).
12. Liao Hung-Jen, Chun-Hung Richard Lin, Ying-Chih Lin, Kuang-Yuan Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications, Vol.36, No.11, (2013), pp.29-45.
13. Liu C., Yang J., Chen R., Zhang Y., Zeng J., Research on immunity-based intrusion detection technology for the Internet of Things, In Proceedings of the 2011 Seventh International Conference on Natural Computation, Shanghai, China, (2011), pp.212–216.

14. Liu L., Xu B., Zhang X., Wu X., An intrusion detection method for internet of things based on suppressed fuzzy clustering, EURASIP Journal on Wireless Communications and Networking, (2018), pp.342-356.
15. Louppe G., Understanding Random Forests, PhD dissertation, University of Liège Faculty of Applied Sciences Department of Electrical Engineering & Computer Science, (2014).
16. Mirjalili S.A., Mirjalili S.M., Lewis A., Grey Wolf Optimizer. Adv in Engi Soft, Vol.69, 2013, pp.46-61.
17. Pahl M.O., Aubet F.X., All eyes on you: distributed multi-dimensional IoT microservice anomalydetection, in: Proceedings of the 2018 Fourteenth International Conference on Network and Service Management (CNSM)(CNSM 2018), (2018), Rome, Italy.
18. Rifkin J., The Zero Marginal Cost Society: The Internet of Things, the Collaborative Commons, and the Eclipse of Capitalism: Book, Apr. (2014).
19. Singh D., Tripathi G., Jara A. J., A survey of Internet-of-Things: Future vision, architecture, challenges and services, IEEE World Forum on Internet of Things (WF-IoT), (2014), pp.287– 292.
20. Tavallaee M., Bagheri E., Lu W., Ghorbani A., A Detailed Analysis of the KDD CUP 99 Data Set, Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), (2009), pp.277-304.
21. Teresa F. Lunt, A survey of intrusion detection techniques, Computers & Security,Vol.12, No.4, (1993), pp.405-418.