# Evaluation of Machine Learning Methods to Detect DoS / DDoS Attacks on IoT

Leonid Galchynsky [1], Mykola Graivoronskyi [1], and Oleh Dmytrenko [1,2]

[1] *National Technical University «Igor Sikorsky Kyiv Polytechnic Institute», 37, Prospekt Peremohy, Kyiv, 03056, Ukraine*
[2] *Institute for Information Recording of National Academy of Sciences of Ukraine, 2, Mykoly Shpaka Street, Kyiv, 03113, Ukraine*

**Abstract**

This paper was aimed at the increase of validity of the chosen method and the accuracy of detection of DDoS attacks on IoT using machine learning. A study of the effectiveness of detecting DoS/DDoS attacks in IoT networks was conducted. The machine learning model was built, the procedure for selecting the most significant 10 features from the dataset data was analyzed, relevant machine learning algorithms were selected to identify traffic anomalies in the IoT network, criteria for evaluating the effectiveness of machine learning algorithms were defined, and performance estimates according to four criteria (time, recall, precision, and f-measure) were obtained. The ranking of the selected and tested machine learning algorithms was obtained by applying the ELECTRE III procedure. As a result, the Random Forest method received the highest rank. The obtained results provide the background for prospective application of the suggested method in real systems for detection of DoS/DDoS attacks in IoT networks.

**Keywords**

Internet of Things, DoS/DDoS attacks, traffic anomalies, machine learning, ELECTRE method

## 1 Introduction

The rapid development of the Internet has given rise to many new phenomena, including the emergence of the Internet of Things (IoT) in various spheres of life. IoT is a set of connected devices, which is controlled by web services or other different types of interfaces [1]. Not surprisingly, this new popular technology has come under the scrutiny of cybercriminals who aim a variety of hacking techniques at IoT systems. This problem is exacerbated by the lack of necessary standards in IoT systems, as well as the specifics of the subject area of IoT: the use of relatively cheap and low-power devices [2]. This specificity makes IoT devices vulnerable to various types of cyberattacks, such as denial of service (DoS) and distributed denial of service (DDoS) [2]. Such attacks tend to cause great damage to entities that use IoT networks, as they easily block the normal operation of devices and, as a result, make it impossible to properly maintain the system. In addition, the technology of these attacks is known for its relative simplicity, which is another aggravating point of the Internet of Things vulnerability problem. Another problem is that traditional high-quality information security solutions are unsuitable for protecting IoT systems. Therefore, the development of this industry requires a relevant system to counter attacks on the Internet of Things, given the high cost of data collected and processed by Internet of Things devices.

Early detection of various vulnerabilities and attacks on IoT devices is extremely important, because devices on the Internet of Things are generally autonomous, i.e. do not require user intervention to work properly. Hence, relevant network security solutions for the IoT system must meet several criteria,

XXI International Scientific and Practical Conference "Information Technologies and Security" (ITS-2021), December 9, 2021, Kyiv, Ukraine
EMAIL: hleonid@gmail.com (L. Galchynsky); mykola.graivoronskyi@gmail.com (M. Graivoronskyi); dmytrenko.o@gmail.com (O. Dmytrenko)
ORCID: 0000-0002-3805-1474 (L. Galchynsky); 0000-0003-2239-2087 (M. Graivoronskyi); 0000-0001-8501-5313 (O. Dmytrenko)

including sufficient speed, accuracy of assessment and a number of other indicators. In our opinion, solutions with machine learning (ML) are one of the most promising for this problem. Although ML information security solutions have gained some popularity, the issue of detecting attacks on IoT networks is still poorly understood.

## 2 Formulation of the problem

Ensuring cybersecurity of IoT networks remains a challenge for professionals because of their features, mainly due to a combination of different technologies, some of which are outdated. These technologies have traditional flaws in data privacy and security, and they need to be addressed to the specifics of IoT. Many researchers are working to address various security issues in the IoT, but the level of security of IoT devices is still not fully in line with user needs. IoT networks are vulnerable to known cyberattacks, including denial of service (DoS), distributed denial of service, replay attacks, man in the middle, routing and eavesdropping attacks [3].

The cybersecurity system should provide network protection at all layers of the Open System Interconnection Reference Model (OSI), providing connection control, analysis of the structure and content of network packets, traffic monitoring, assessment of system elements functioning states [4].

However, the current practice shows that the analysis of security threats and methods of their detection are limited to one layer of the network interaction while for IoT networks, real attack can be a combination of threats from multiple network layers. Standard protection techniques developed for certain layers of IoT and sensor networks may be ineffective against attacks from other layers. It follows that a reliable assessment requires an analysis of the network as a whole. However, solving this systemic problem in general is too complex to be completed in one study. Therefore, we will consider only one of the biggest cyber threats to IoT networks – DoS/DDoS attacks. The main danger of the vast majority of DoS/DDoS attacks – in their absolute clarity and almost "normal" behavior. Software errors, when discovered, are immediately fixed and promptly corrected. But in this case the only apparent indication of the attack is the full consumption of resources, what is almost normal behavior for modern information systems. Standard methods of statistical analysis do not allow us to detect previously unknown attacks, and hence machine learning algorithms become the tool to solve this problem which is actively studied and used [5]. As the analysis shows, the most effective means to detect and prevent DoS/DDoS attacks are those that use intelligent methods (IDMS) [6]. In recent years, many studies have been conducted on the use of machine learning as a tool to detect DDoS attacks. However, they are often not sufficiently substantiated, there are few details of the results obtained, and they also rarely specialize in IoT networks, and evaluate the results only by accuracy, or do not clearly describe the approach by which the solution was chosen.

## 3 Analysis of recent research and publications

The objective of detecting DDoS attacks consists in making the most reliable estimate of whether devices are under attack in a given period of time or not by analyzing data from devices. The speed of the chosen method is the most desirable, because the slightest delay can lead to significant losses. Standard methods of statistical analysis do not allow to detect previously unknown attacks, therefore the appropriate tool to solve this problem is to use machine learning algorithms [7]. Such a strategy for detecting DoS/DDoS attacks is implemented through intrusion detection systems (IDS). Network Intrusion Detection Systems (NIDS) are usually located on the node that connects the internal network to the Internet, in order to scan incoming and outgoing traffic for known attack patterns [8]. As known malicious patterns are detected, messages will be displayed and examination algorithms may be run, which reduces response time and potentially increases the accuracy of the investigation. This is why IDS should include ML in its anomaly detection process with classification, clustering, and other methods used to detect anomalous traffic.

To create a prototype of a real ML-based intrusion detection system, it was necessary to develop a software application and to test it on the appropriate data set. For the experiments, we chose the BotIoT data set, created in the laboratory of the Australian Center for Cyber Security (ACCS) specifically for researchers to study the possibilities of machine learning in detecting attacks on IoT networks.

Due to the limited power consumption of IoT devices, the key management system may be compromised. Simplício et al have shown that simplified key authentication systems are insufficiently reliable in high-scale implementations [9]. Network protocols such as IPv6 and IPv4 are targets for attempting to establish remote access. Czyz et al demonstrated that IoT devices could be accessed remotely via the command line interface using the Telnet protocol [10]. IoT networks often process big data continuously. Loss of data or denial of service can lead to huge amounts of network traffic and loss of control over the system. According to Angrishi [11], malicious software can remain dormant in IoT devices, and after launch, it turns the IoT device into a bot and uses it to carry out a DDoS attack.

In 2013, the Austrian and German power grids began to fail due to a DDoS attack and filled the central command center with traffic [11]. A DDoS attack can temporarily disable the communication network and IoT device protocol. The automated smart grid system relies heavily on the IoT monitoring network. The sensor network collects real-time information data to monitor the condition of equipment and control systems. As a result, in the event of an attack, the station will lose the ability to monitor the transmission and distribution network.

**The aim of this paper** is to increase the validity of the chosen method and the accuracy of detection of DDoS attacks on IoT using machine learning.

# 4 Results and discussion

## 4.1 The formal statement of the problem

The formal statement of the problem of detecting DDoS attacks is as follows:

let a certain set of input data **X** and a set of responses **Y** exist, where $x_i \subset \mathbf{X}$ is a set of characteristics of traffic data, $y_i \subset \mathbf{Y}$ determines the affiliation of a particular $x_i \subset \mathbf{X}$ to the attack.

We assume there is a training sample $\{ x_1, \dots, x_n \} \subset \mathbf{X}$ and a corresponding sample of correct responses $\{ y_1, \dots, y_n \} \subset \mathbf{Y}$.

The problem of detecting an attack in this case may be formulated as the definition of such a rule (algorithm or model), which will give the value of **Y** closest to the correct values on the whole set **X**:

$a$: $X \rightarrow Y$

The characteristics $x_i \subset \mathbf{X}$ are divided into types:

- Quantitative (values from the set of real numbers);
- Boolean {0, 1};
- Nominal (values from a finite subset of N);
- Ordinal, representing nominal characteristics, with a linear order.

A Boolean characteristic can be a column of data that indicates the presence or absence of an attack. The protocol to which this package belongs can be a nominal characteristic. The number of flags in the package headers can be an ordinal characteristic. An example of quantitative characteristics is the package length.

The training sample is created from a specific data set in the form of a feature matrix, in which the rows represent examples of traffic, as well as the corresponding known states.

**Y** can also consist of features of different types: Boolean, nominal, intersecting nominal, and example data can belong to several types of attacks simultaneously. The peculiarity of the last two options is that first it should be indicated whether the example is an attack, and only then may its type be determined.

## 4.2 Two approaches to solve the problem

Two approaches are proposed to solve this problem:

- detection of attacks based on signatures;
- detection of attacks based on anomalies.

Signature-based identification is commonly used to identify known types of attacks. No description of typical actions is required, but a database with known attack signatures is required to detect these types of attacks. This is the main disadvantage of this approach. However, even for known signatures,

there may be another vulnerability: if there are too many signatures, searching the database can take too long to detect intrusions in a timely manner.

Anomaly-based intrusion detection methods recognize unusual activity in network traffic. For example: abuse of system rules (hiding the IP address interval, performing a standard transaction on a hidden port); unusual traffic patterns (more UDP packets than TCP packets); suspicious examples in the data section of the program, etc.

The most difficult problem for anomaly-based detection methods is the detection of atypical system behavior, as well as the choice of boundaries to prevent false alarms. Statistical methods of analysis do not allow to detect previously unknown attacks. The usual way to create a model that has the ability to avoid a lot of false positives is extremely difficult. Therefore, in fact, the only way out is to use machine learning tools to detect various threats [12].

## 4.3 Construction of a machine learning model

Constructing a model of machine learning involves the following steps:

1. Preparation and collection of data;
2. Selection of characteristics (parameters);
3. Selection of the set of algorithms;
4. Choice of criteria and factors;
5. Training;
6. Evaluation of the effectiveness of the results.

The scheme of step-by-step construction of machine learning model is shown in Fig.1.

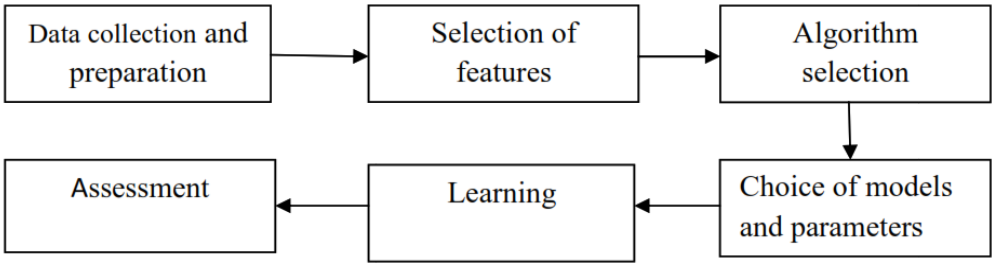


**Figure 1**: Stages of construction of the ML model

**The first stage** involves filtering and normalizing the data to provide input data for algorithms and characteristics. Those that are important for the learning process are selected. Data may contain noise, may not have a clear structure, so they must be pre-processed. Good data preparation allows for effective analysis, limits errors and inaccuracies that may occur with the data during processing, and makes all processed data easy to use.

**At the second stage**, among all the existing characteristics, those that are relevant to the learning process are selected. The number of characteristics in the data sets can reach hundreds. And not all of them are really important (i.e. have a relationship with the target variable), so they actually are redundant (or noise). Deleting such parameters helps the researcher to better understand the data, as well as reduces the time to set up the model and the accuracy of its forecasts.

**At the third stage**, relevant algorithms are selected to solve a specific problem. Analysis of publications [5], [13] allowed to choose five algorithms that are most suitable for detecting anomalies in traffic. These include:

- The method of k-nearest neighbors (KNN);
- Naive Bayesian Classifier (NB);
- Random Forest;
- Logistic Regression;
- Decision Tree.

**The fourth stage** involves the selection and justification of criteria by which one can assess the quality of the algorithm in terms of the problem, as well as the optimal parameters of algorithms to solve the problem are selected.

**At the fifth stage**, the model is taught using part of the data set as training data. Part of the data is taken from the whole set and given to the algorithm to detect patterns; afterwards it should be able to make assumptions based on the data obtained in this stage.

**Finally, at the sixth stage**, the model is tested on data other than the training ones in order to find out the quality of the forecast.

## 4.4 Creation of a working prototype

To create a prototype of a real ML-based intrusion detection system, it was necessary to develop a software application and test the appropriate data set. For the experiments, we chose the BotIoT data set, created in the laboratory of the Australian Center for Cyber Security (ACCS) [14] specifically for researchers to study the possibilities of machine learning in detecting attacks on IoT networks.

The reasons for this choice are as follows: it is open, it contains traffic captured from the IoT network, it has samples of a wide variety of attacks, including a large number of examples of DoS/DDoS attacks using different protocols, it includes real traffic, and provides the possibility to generate new characteristics from the raw data set. The published recorded .pcap files are 69.3 GB in size and contain over 72,000,000 entries. However, so far we have used only 5% of the original data set retrieved using MySQL queries. This extracted 5% consists of four files with a total size of about 1.07 GB and includes about 3 million records. This set is known as Bot-IoT. The data set includes real and simulated IoT network traffic along with different types of attacks [15]. Attacks in it are divided into three types: probing attacks, DoS and information theft. However, it also needs careful handling. Figure 2 provides a diagram of the distribution of the number of packages by category. In this diagram, we can see a significant uneven amount of data generated during DoS/DDoS attacks compared to other attacks and normal data. On the one hand, this indicates an imbalance in the data in the dataset, but on the other hand, it reflects the real situation.

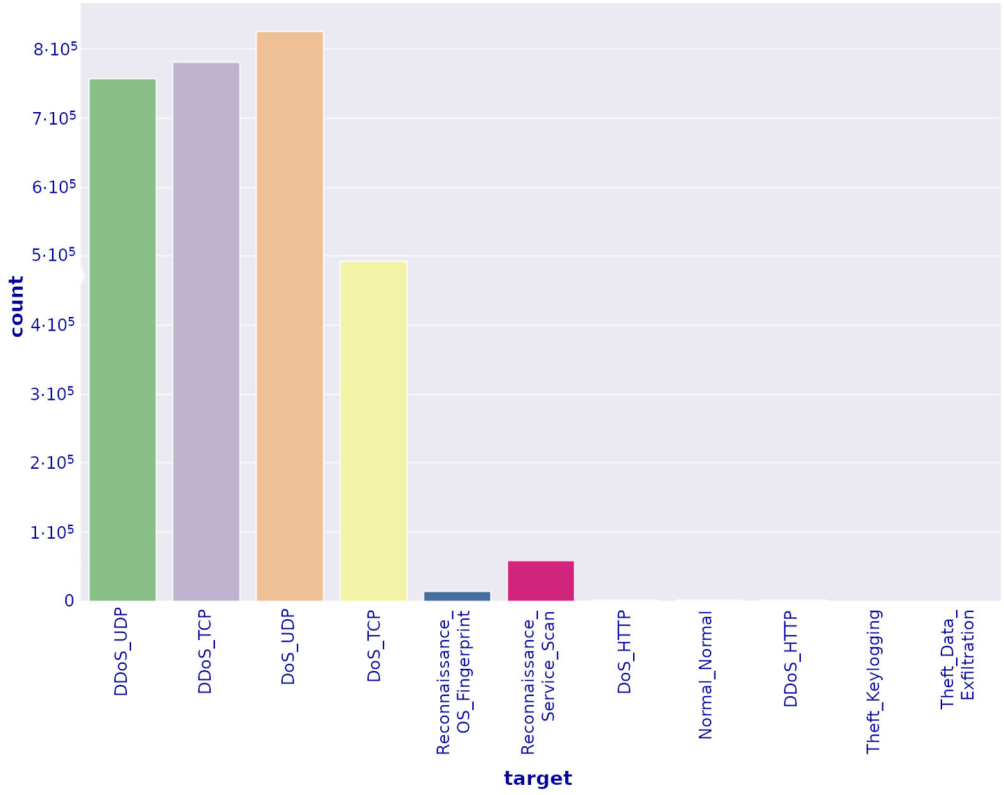


**Figure 2:** Number of packets by categories

In the data set, the instances of the attack are marked "1", while the usual traffic "0" for learning and testing machine learning models using binary code. In addition, attributes of the attack category and subcategory were introduced that could be used to train and test multiclass classification models.

Then a two-stage procedure of selection of significant characteristics was carried out. This process was designed primarily to improve the predictability of classifiers. First, the selection from raw network traffic data was made using the CICFlowMeter tool [16]. CICFlowMeter reads a pcap file and creates a visual representation of the removed functions, as well as generates csv files for the data set.

The selection of features in this software product is based on the Pearson correlation matrix, and in addition to the already specified characteristics, there is a procedure for generating aggregate characteristics. 14 characteristics were obtained in the first stage of 33 characteristics available in the data set. And then the procedure of selection of characteristics on value of Ginny's impurity was carried out to make further selection from these 14 characteristics. In the end, 10 significant characteristics were finally selected, according to the combined assessment presented in Table 1.

To ensure correct assessment of the quality of classifiers and to avoid the problem of overfitting, the data were divided into training and test parts.

Models learn only from the data of the training data set. Even when using cross-validation to determine the optimal parameters of the model, cross-validation is performed only from training data.

There are no elements from the training part in the final data set. Thus, the model we derived by performing learning from training data does not know the exact answer to any of the elements of the final sample and will make predictions based on the rules and patterns developed during training.

In our study, we divided the original data set at 60% for the training sample and 40% for the test sample. This gives a fairly even distribution, so the large size of the training sample reduces the likelihood of underfitting when the model lacks data for training. To better divide each class into training and test samples, data shuffling was used when sampling the random_state parameter to ten.

**Table 1**
Significant characteristics

| Characteristic | Description |
|---|---|
| state number | Numerical representation of the state of characteristics |
| seq | Argus sequence number |
| N_IN_Conn_P_SrcIP | The number of incoming connections per the source IP address |
| mean | Average duration of the aggregated records |
| stddev | Standard deviation of the aggregated records |
| N_IN_Conn_P_DstIP | The number of incoming connections per the destination IP address |
| min | Minimum duration of the aggregated records |
| max | Maximum duration of the aggregated records |
| srate | Packets from the source per second |
| drate | Packets to the destination per second |

After data analysis in BoT-IoT the following columns were coded: saddr, daddr, proto, target. Missing values were replaced by NaN, we also left only columns with the code:

- DDoS_HTTP – 0,
- DDoS_TCP – 1,
- DDoS_UDP – 2,
- DoS_HTTP – 3,
- DoS_TCP – 4,

- DoS_UDP – 5,
- Normal_Normal – 6.

Next, we conducted a series of tests to evaluate the effectiveness of anomaly-based detection of DoS/DDoS attacks. The machine learning algorithms defined above were tested using the described data set. The description of the selected algorithms is detailed in the literature. The experiments presented in this work were performed on a Dell Inspiron 15 3000, with Windows 10 operating system with AMD A6-6310 processor at 1.80 GHz, and 4 GB memory. For the software implementation, scikit-learn, Mat-plotlib, Pandas, and NumPy machine learning libraries written for the Python 3.9 programming language, which are open source and widely used in modern machine learning research, were used.

Before testing ML algorithms, it is important to determine which indicators to evaluate. The most important performance indicators are as follows: prediction time, precision, recall, accuracy, and *f*-measure as shown in the equations below [14]:

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

$$f - measure = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} \tag{4}$$

where *TP* is the number of true negative results, *FN* is false negative, *TN* is true negative, and *FP* is false positive. In particular, the *f*-measure is indicative, because it reduces two other fundamental metrics, accuracy and completeness, to one number.

Precision is interpreted as part of the objects called positive by the classifier which are really positive, and recall shows what part of positive class objects of all positive class objects was found by the algorithm. The very introduction of precision does not allow us to write all objects into one class, because in this case we get an increase in the level of False Positive. Thus, recall demonstrates the ability of the algorithm to detect the positive class in general, and precision – the ability to distinguish this class from other classes. Accuracy metrics are not very useful in problems with unequal object classes [14]. Precision and recall do not depend, in contrast to accuracy, on the class ratio and are therefore applicable in unbalanced samples.

To implement the learning phase, a program was written using built-in functions in the sklearn library. To prepare the algorithms and select the optimal parameters, the implementation of algorithms in the sklearn library was used.

For training, we selected random samples of data that were more balanced. Figure 3 shows a screenshot of the data distribution chart by category.

Next, experimental tests of the ability to determine anomalies for each trained model were performed 100 times on the test set of the data set. To average the metrics, this procedure was repeated 10 times.

All data collected for testing algorithms on the BoT-IoT data set are presented in the table. As you can see from the Table 2, all algorithms showed very high results in detecting instances of DoS attacks via HTTP protocol. Deviations in the results are insignificant, so to detect this attack in real applications using the fastest option could be recommended. However, such conclusion could be made at first glance only. In fact, the purpose of this study is to obtain a sounder solution for the different types of DoS and DDoS attacks. For example, based on the data obtained in the case of logistic regression, naive Bayesian classifier and decision tree give a significantly lower percentage of detection of normal traffic, which means an increased risk of false positives.


**Figure 3**: Sampling of data by categories: 0 – DDoS_HTTP, 1 — DdoS_TCP, 2 — DdoS_UDP, 3 — DoS_HTTP, 4 — DoS_TCP, 5 — DoS_UDP, 6 — Normal

By analyzing the time spent on obtaining the forecast, one can find a clear "outsider" – the KNN method, which is ten times slower than the random forest, which in turn is slower than the rest. For Logistic Regression, the decision tree, and the naive Bayesian classifier, the time spent on 100 predictions is comparable and is within 1.5 seconds. In this case, without an integrated assessment of the criteria, it is difficult to determine which of the algorithms best coped with the task, and which ratio of time and accuracy can be considered optimal.

## 4.5 Choice of the best algorithm

In order to obtain sound estimates of the quality of machine learning algorithms in the detection of DoS/DDoS attacks, it is necessary to set the task of objectively choosing the best option from several, that is to set and solve the problem of decision making. The results of evaluating the selected algorithms using the selected dataset on these five metrics did not show a clear advantage of any of them. For a reasonable choice of the best, it is necessary to apply a method that takes into account the multi-criteria nature of such a choice. In a situation of multi-criteria choice, it is not clear which solution is better, so you need to find a compromise solution that takes into account the importance of each criterion. This leads to the concept of effective (Pareto optimal) solution. The property of efficiency (in extreme cases, poor efficiency) must have any solution that claims to be called the best.

Since there is no general solution to the Pareto optimal choice problem, the decision support methodology offers many methods, from the set of which one should choose the one that would correspond to the task. Among the many methods of multi-criteria selection, two classes can be distinguished – methods that focus on the utility function, and methods that focus on pairwise comparison of criteria with the subsequent matching procedure. In our case, it is advisable to rely on the latter approach, because the generally accepted utility function for this task does not yet exist.

One of the leaders of this approach is the method ELimination Et Choix Traduisant la REalite (ELECTRE) [17]. This method is characterized by four aspects of advantage modeling: modeling of fluctuations (partial and comprehensive), modeling of comprehensive incomparabilities, the concept of consistency and relative importance of criteria, and the concept of inconsistency and veto thresholds. A number of factors influenced the specific choice of the ELECTRE III method for the algorithm rating problem. Second, ELECTRE was designed for the fuzzy (inaccurate and uncertain) nature of decision-making, using thresholds of indifference and preference.

**Table 2**
The results of the evaluation of algorithms

| Criterion | | KNN | Random Forest | Logistic Regression | NB | Decision Tree |
|---|---|---|---|---|---|---|
| Time for 100 predictions | | 73,1 | 2,87 | 1,5 | 1,7 | 1,45 |
| DoS UDP | f-measure | 0,99 | 1 | 0,04 | 0,5 | 0,94 |
| | recall | 0,99 | 1 | 0,66 | 0,96 | 0,98 |
| | precision | 0,98 | 1 | 0,02 | 0,34 | 0,9 |
| DDoS UDP | f-measure | 0,986 | 0,983 | 0,99 | 0,817 | 0,967 |
| | recall | 0,99 | 0,988 | 1 | 0,77 | 0,975 |
| | precision | 0,982 | 0,979 | 0,99 | 0,87 | 0,96 |
| DDoS TCP | f-measure | 0,99 | 1 | 0,96 | 0,65 | 0,94 |
| | recall | 0,98 | 1 | 0,93 | 0,53 | 0,9 |
| | precision | 0,99 | 1 | 0,99 | 0,85 | 0,91 |
| DoS TCP | f-measure | 0,97 | 0,983 | 0,99 | 0,92 | 0,99 |
| | recall | 0,98 | 0,988 | 0,99 | 0,84 | 0,99 |
| | precision | 0,972 | 0,979 | 1 | 0,99 | 0,98 |
| DoS HTTP | f-measure | 0,98 | 0,994 | 0,985 | 0,999 | 0,962 |
| | recall | 0,98 | 0,991 | 0,99 | 0,999 | 0,953 |
| | precision | 0,98 | 0,998 | 0,98 | 0,999 | 0,971 |
| DDoS HTTP | f-measure | 0,964 | 1 | 0,545 | 0,364 | 0,726 |
| | recall | 0,99 | 1 | 0,67 | 0,27 | 0,57 |
| | precision | 0,94 | 1 | 0,46 | 0,56 | 0,998 |
| Normal | f-measure | 0,979 | 0,989 | 0,758 | 0,618 | 0,853 |
| | recall | 0,96 | 0,99 | 0,702 | 0,46 | 0,901 |
| | precision | 0,998 | 0,999 | 0,823 | 0,95 | 0,81 |

Another feature of ELECTRE, which distinguishes it from many criteria of solution methods, is that it is fundamentally non-compensatory. This means, in particular, that a very poor score on a criterion cannot be offset by a good score on other criteria. The original feature is that the ELECTRE models allow for incomparability. Incomparability, not to be confused with indifference, arises between any of

the alternatives a and b, when there is no clear evidence in favor of which of the alternatives to prefer. This is the situation we are faced with, choosing the best: speed with more errors, or vice versa.

The purpose of applying ELECTRE methods is to narrow the Pareto set of alternative solutions. This is done as follows: for each of the criteria (it is assumed that they are numerical) the "weight" is determined by the results of the survey – a number that characterizes the importance of the criterion. In all modifications of the ELECTRE method, an attempt is made to obtain qualitative information on the relative importance of the criteria (statements such as "criteria 3 and 4 are of equal importance and considered together are more important than criterion 1") and convert it into quantitative, numerical. In all modifications of the ELECTRE method, in the first stage the weights of the criteria are determined with the help of a decision-maker – positive real numbers, which are the greater, the more the corresponding criterion is important for the decision-maker. This approach, of course, has a drawback – the estimated determination of weights. However, it is impossible to completely avoid subjective assessments in the decision-making process, it is only necessary to approach the determination of weight with great care. In addition, these assessments should be provided by experts, not in general, but in individual aspects.

The ELECTRE III method involves the following steps – calculating the concordance matrix; calculating the divergence matrix; calculation of the reliability matrix; calculation of the matrix of preferences. In the last matrix is the resulting rating for each of the alternatives.

Evaluations from five experts provided information on the relative importance of the criteria. All experts determined that the detection of all subcategories of DoS/DDoS attacks is equally important, and the correct work in detecting normal traffic is more important than individual categories of attacks. Averaging the estimates, we obtain the following weights: time – 0.8, detection of normal traffic – 2, detection of denial-of-service attacks – 6 (1 for each criterion). In addition, a survey was conducted to determine three thresholds, preference (p), indifference (q), and veto (v).

To calculate the results, the XLSSTAT package was used, which provides a user interface for applying the ELECTRE II and ELECTRE III methods. To evaluate the performance of algorithms, the f-measure, as an integral indicator of recall and precision, as well as the time for which the algorithm will perform estimates, were chosen as criteria. The f-measure value was calculated, and the time was converted into a dimensionless value in the interval (0; 1], where 1 is the best value available (the fastest result among the five selected algorithms).

As a result of calculations, omitting intermediate values, we obtain a matrix of preferences in the terminology of ELECTRE III (Table 3).

**Table 3**
The matrix of preferences

| | Random Forest | Decision Tree | Logistic Regression | KNN | NB |
|---|---|---|---|---|---|
| Random Forest | I | P | P | P | P |
| Decision Tree | NP | I | P | P | P |
| Logistic Regression | NP | NP | I | P | P |
| KNN | NP | NP | NP | I | P |
| NB | NP | NP | NP | NP | I |

The letter P (preference) in the corresponding cell means the superiority of the algorithm over another algorithm, and NP (non-preference) – vice versa. This means that the ELECTRE III method allowed to integrally evaluate the effectiveness of the tested machine learning algorithms for detecting DoS/DDoS attacks. Random Forest is a better choice than 4 other algorithms, Decision Tree is better than other alternatives, Logistic Regression is better than two algorithms, KNN is better than one algorithm. NB was last in the rankings.

## 5 Conclusions

A study of the effectiveness of detecting DoS/DDoS attacks in IoT networks was conducted. To accomplish this task, a machine learning model was built, in which a set of data was selected (BoT-IoT dataset), the procedure for selecting the most significant 10 features from the dataset data was analyzed, relevant machine learning algorithms were analyzed and selected (Random Forest, Decision Tree, Logistic Regression, KNN, NB) to identify traffic anomalies in the IoT network, criteria for evaluating the effectiveness of machine learning algorithms were defined, training on a selected set of datasets was conducted, efficiency on a test sample of the dataset was evaluated, and performance estimates on four criteria (time, recall, precision, and f-measure) were obtained. The ranking of the selected and tested machine learning algorithms was obtained by applying the ELECTRE III procedure, designed for multi-criteria selection, which focuses on a pairwise comparison of criteria with the subsequent matching procedure. As a result, the Random Forest method received the highest rank. The obtained results give perspective for application in real systems for detection of DoS/DDoS attacks in IoT networks. This study cannot be considered complete and calls for further work.

**The prospect for further research** is to refine the presented model by using a larger set of data for training and testing, expanding the range of potentially suitable machine learning algorithms, obtaining estimates of the effectiveness of algorithms by alternative methods.

## References


[1] N. Moustafa, B. Turnbull, K.-K.R. Choo, Towards automation of vulnerability and exploitation identification in iiot networks, in: 2018 IEEE International Conference on Industrial Internet. DOI:10.1109/ICII.2018.00023 Corpus ID: 53948805

[2] N. Moustafa, B. Turnbull, K.-K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of IoT, IEEE Internet Things J. (2018). DOI:10.1109/ICII.2018.00023

[3] P. Kasinathan, C. Pastrone, M.A. Spirito and M. Vinkovits, Denial of service detection in 6LoWPAN based internet of things, in: IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2013, pp. 600-607. DOI:10.1109/WiMOB.2013.6673419

[4] S. Salnyk, A. Storchak, A. Mykytyuk, Information Technology and Security 7 (2019), Iss. 1, pp. 25-34. URL: http://nbuv.gov.ua/UJRN/inftech_2019_7_1_5

[5] M. Tayyab, B. Belaton, and M. Anbar, "ICMPv6-based DoS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: a review," IEEE Access 8 (2020), pp. 170529–170547. DOI: 10.1109/ACCESS.2020.3022963

[6] Ahmed, Sheikh, A Study of ML Algorithms for DDoS Detection, International Journal for Research in Applied Science and Engineering Technology (2021).

[7] Y. Alshboul, and K. Streff, "Analyzing Information Security Model for Small-Medium Sized Businesses", in Proc. 21st Americas Conference on Information Systems, Puerto Rico, 2015.

[8] DoS and DDoS vulnerability of IoT: A review, URL: https:// www.researchgate.net/publication/39862422_DoS_and_DDoS_vulnerability_of_IoT_A_review.

[9] M. A. Simplício, M. V. M. Silva, R. C. A. Alves, and T. K. C. Shibata, "Lightweight and escrow-less authenticated key agreement for the internet of things", Comput. Commun., 98 (2017), pp. 43–51.

[10] J. Czyz, M. J. Luckie, M. Allman, and M. Bailey, "Don't forget to lock the back door! a characterization of ipv6 network security policy", in NDSS, 2016.

[11] K. Angrishi, "Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets", CoRR, vol. abs/1702.03681, 2017. arXiv: 1702.03681. URL: http://arxiv.org/abs/1702.03681.

[12] D. Parwani, A. Dutta, P. Kumar Shukla, et al, Various techniques of DDoS attacks detection and prevention at cloud: a survey, Orient. J. Comp. Sci. and Technol. 8 (2015), no. 2, URL: http://www.computerscijournal.org/?p=1983
[13] Jadel Alsemiri & Khalid Alsubhi, Internet of Things Cyber Attacks Detection using ML, International Journal of Advanced Comp. Science and Applications (2019).
[14] Nour Moustafa, Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic, PhD thesis, University of New South Wales, Canberra, Australia.
[15] K. Nickolaos, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the IoT for network forensic analytics: Bot-iot dataset", Future Generation Comp. Systems 100 (2019).
[16] Mehryar Mohri, Afshin Rostamizadeh, & Ameet Talwalkar, Foundations of Machine Learning, 2nd. ed., the MIT Press, 2018.
[17] Jose Figueira, Salvatore Greco, Bernard Roy, Roman Slowinski, Electre Methods: Main Features and Recent Developments, 2010. hal-00876980