Random Forest Detector and Classifier of Multiple IoT-based DDoS Attacks

VANYA IVANOVA, TASHO TASHEV, IVO DRAGANOV
PhD School, French Faculty of Electrical Engineering
Technical University of Sofia
8 Kliment Ohridski Blvd., 1756 Sofia
BULGARIA

Abstract: - In this paper two new models for Random Forest (RF) classifiers are presented. The first one discriminates Distributed Denial of Service (DDoS) network attacks from normal IP (Internet Protocol) traffic and the second one classifies 10 types of attacks. General optimization procedures are proposed based on the parameters of the RF classifiers. The observed DDoS attacks are typical for botnets, comprised of IoT (Internet of Things) devices. Bot-master plays central role into coordinating the bots. The explicit aim is either resource exhaustion of the targeted machine or bandwidth saturation of the supporting channels to it. Both activities render the legitimate services unavailable. The detection process has an accuracy of 0.9999. The classification process deviates between 0.9992 and 0.9999. Processing times allow the proposed approach to be used in real-world applications.

Key-Words: - IoT, DDoS, network, normal traffic, malicious traffic, detector, classifier, Random Forest

Received: March 15, 2021. Revised: January 20, 2022. Accepted: March 12, 2022. Published: April 13, 2022.

1 Introduction

Information services, both in the Software Defined Networks (SDN) and in Cloud Environment, as well as in general purpose networks, has been deeply affected by Distributed Denial of Service (DDoS) attacks over the years [1]. In many instances of such attacks IoT devices have been engaged as bots, or malicious nodes, to carry out flooding over the target machine over multiple paths and thus render the offered services inaccessible. Researchers have been trying to limit their negative effect by introducing various predictors, analyzing the network traffic, using high level features and applying different techniques from the machine learning field [2].

Random Forest (RF) is one of the algorithms, extensively used for this particular task. Hypertext Transfer Protocol (HTTP) DDoS attacks, as one of the most common ones, is an object of study of Idhammad et al. [3], where the Low and Slow, as well as the typical flood attack, have been considered. Combining Information Theoretic Entropy with RF, the authors developed detection system for the cloud environment. Classification accuracy is being reported as high as 99.54%, with the False Positive Rate (FPR) as low as 0.4% and processing time of about 18.5 sec. Another example of the application of the RF algorithm is the detection of Domain Name System (DNS) DDoS attacks [4]. Traffic classification within this study led to 99.2% accuracy and the developed model is considered applicable over large-scale query flows. Execution time is also thought to allow real-time operation. As alternative approach to the direct discovery of the DDoS attack itself, Lu et al. [5] propose the detection of the Command and Control (C&C) session detection by application of RF. This technique allows to early spot a sign of forthcoming DDoS attack by preparing the bots to accomplish it by exchanging data with them from the so called C&C server. A feature vector, comprised of 55 elements from the network traffic, is feeding the input of an RF classifier, which not only achieves 99.05% classification accuracy and just 1.23% false alarms rate, but also is capable of better high-dimensional training and finds the importance of the various features. It turns out that this classifier is more efficient as a solution to this problem than the Support Vector Machine (SVM) and the Naïve Bayes (NB). Aiming specifically on the IoT-based DDoS attacks, Farukee et al. [6] apply the RF as a feature selector, while also using the deep learning as a major technique for classification. Classification accuracy of 99.63% is achieved for the 1D-CNN (Convolutional Neural Network) and 99.58% - for the MLP (Multilayer Perceptron) over the test set with network samples, consisting the attacks. Another attempt to propose an efficient

measure against the pernicious influence of C&C mechanism over the network availability when saturated with unwanted traffic by the DDoS attack is presented in [7] by Pande et al. They also consider issues, related to confidentiality and integrity of data being exchanged over the network in such situations. Discrimination of normal vs. attack samples tends to be successful at a rate of 99.76%.

Behavioral analysis of the network flows are being done also by comparing the RF algorithm with the Dense Neural Networks [8]. It is shown that, although they could perform almost equally well, the RF classifier demands less samples, achieving satisfactory accuracy. RF reaches a Precision between 0.87 and 1.0, Recall between 0.56 and 1.00 and F1-score between 0.68 and 1.00 over 7 types of anomalies, coming from DDoS attacks. In [9] it is demonstrated that the Random Forest classifier takes the leading position under an unified scale with a ranking of 8.50, followed by the Decision Tree algorithm with 7.25, k-NN (k-Nearest Neighbors) – with 6.75, the MLP – with 2.75 and lastly the Naïve Bayes – with 2.25, when compared by efficiency of solving the DDoS detection task. The overall accuracy of the RF is 98.9%, with a Precision of 99.9 for the normal samples and 99.6% for the DDoS ones. Experimentation with a more diverse dataset [10], including samples from 11 DDoS attacks, reveal slightly lower classification efficiency for the RF algorithm, but still it takes the leading position in front of Naïve Bayes and Logistic Regression. It gets Precision of 0.77, 0.56 for Recall and 0.62 – for the F1-measure. Only the ID3 algorithm achieved slightly higher values – 0.78, 0.65 and 0.69, respectively.

Early detection of DDoS attacks and their further mitigation from a practical standpoint has been proposed in [11]. A Ryu framework and the RF algorithm are the base of an implementation, used for SDNs. Taking into account the flow entries, the RF classifier distinguishes the normal and attack packets and for the latter additional rules are imposed to the switches in order to limit their flow. Average accuracy of 98.38% is being reported with an average detection time of 36 ms. The mitigation system itself needs 1179 ms in order to lend an effect of reducing the attack, during which process a sparing of 44.9% of usage of the targeted machine CPU has been measured. The *nmeta2* traffic classification platform is another example of a tool for SDN based study of the efficiency of various machine learning algorithms towards the discovery of DDoS attacks [12]. With its help the following F1-measures are being registered over a network dataset, containing 8 features: RF – 0.9548, k-NN – 0.9484, and SVM – 0.9311. RF and the k-NN turns out to be very close in initialization times with the SVM far away behind them, more than 74 times slower. In another comparison [13], the RF shows almost equal Precision with the NB and MLP about the detection of normal traffic and UDP flood (between 0.95 and 0.99). It was less effective in discovering Smurf attacks (just above 0.5), compared to the MLP (close to 1.0), but close to it when spotting SIDDOS (SQL (Structured Query Language) Injection DDoS) and HTTP-FLOOD (around 0.85 and 0.93, respectively). In the latter 3 cases, the NB has significantly lower Precision. With the exception of the Smurf attacks, all 3 classifiers show similar Recall, varying between 0.9 (for the SIDDOS) and almost 1.0 (for the normal traffic). For the Smurf attacks the Recall for RF and MLP are as low as 0.3 with the NB going down to almost 0. Obviously, extending the range of detectable attacks would require a combined type of a classifier, rather than a single one, undergoing prolonged training. Blockchain IoT based systems are also affected by the DDoS attacks and fog computing has been applied in order to limit their influence on the basis of distributed framework [14]. Threats on the smart contracts, as revealed in 2016 and 2017 with massive types of decentralized attacks, firmly point out the need of protection of this systems as well, which currently does not exist as a complete solution. RF and XGBoost are thought to be two of the perspective machine learning techniques for discovering of such threats. Effectiveness of the protection process is being sought into the introduction of the interplanetary file system as addition. Fog computing is the mean, which Kumar et al. [14], propose for implementing a distributed framework, incorporating these 2 mechanisms. RF achieves detection rate as high as 99.99%, using 10 features, aggregated from simulated network data flows. Reducing the computational burden of the discovery process of DDoS attacks is another aspect of the research in this field, which plays a crucial role into the practical application of developed methods. In [15], Ustebay et al. propose a reduction of the features by a recursion, using the RF algorithm and also a deep learning technique, while constructing an Intrusion Detection System (IDS). The introduction of importance value of the features ease this process and leads to intrusion detection of 91% from the generated features, being further processed by a Deep MLP (DMLP).

The RF algorithm is being implemented in enhanced version in order to get a mobile

application for detecting DDoS attacks as shown by Prasad et al. [16]. The authors use that technique to get in-app notifications about forthcoming attacks and to block certain IP addresses, posing a threat. Continuous traffic analysis by the RF algorithm led to 95.19% Accuracy with 95.10% Precision and 94.47% F1-measure. The accuracy of the k-NN and the Decision Tree (DT) algorithms in the same time is 87,34% and 93.83%, respectively. Subsequent tuning of the RF algorithm leads to 97% Accuracy, using 15 features from 80 initially aggregated from the network flow. Another path towards obtaining higher levels of Accuracy into detecting DDoS attacks is proposed by Nandi et al. [17] by the introduction of a hybrid feature selection approach. It includes several criteria, related to various statistical significance parameters (Information Gain, ReliefF and others) based on contained information in the features, and then applying few classifiers. Using this hybridly selected features the RF algorithm achieves 99.86% Detection Rate, followed by the J48 algorithm with 99.79% and then the Decision Table algorithm with 99.48%. Multiagent Intrusion Detection Systems (IDS) are another approach against the spreading of DDoS attacks, used for securing IoT networks in particular [18]. Combining J48 algorithm with multidirectional selection of features, yielding the highest Information Gain, leads to especially efficient Detection Rate of such attacks. When varying the number of features between 15 and 56, that combination reaches Detection Rate of up to 0.998, which is also reached by the RF algorithm in most of the cases, and followed by the NB algorithm with as high result as 0.965. The most informative selection of features is also primary aim of the study of Gaur and Kumar [19]. They use chi-square, Extra Tree and ANOVA methods for feature selecting, which are then passed to the RF, DT, k-NN and the XGBoost classifiers. Feature reduction rate has been reported as 82.5% and leading to accuracy of 98.34% for the XGBoost with ANOVA as selection criterion. The data samples are gathered from attacks aimed at IoT devices. Gaussian Mixture Models (GMM) and Universal Background Model (UBM) are relatively new approaches in the field of DDoS attacks detection, successfully applied by Osorio et al. [20]. RF is also tried over the same, real-world, dataset. It has an accuracy varying between 85.13% and 96.7%, while the UBM has it between 60% and 77.5%, and the GMM – 80.3% on average.

In this study, the main aim is to find the optimal configuration of the RF algorithm in its basic form, applied over freely available and commonly used IoT-Bot dataset [21], once as a detector of IoT-based DDoS attacks and secondly – as a classifier for each type, out of 10 attack types, so it could be further used either as a standalone implementation , or as a part of combined classifier. Experimentation has been done over the full set of 10 features, originally selected in the dataset, and once more over a reduced set of 8 of them, corresponding to the most informative content of the dataset to the dominant part of the attacks.

In the Section 2 of the paper a brief description of the dataset is given with the types of attacks in it, associated features, and then their mutual distribution by attack is presented, and also rearranged in space using the PCA [22] transformation method. Then, description of the RF algorithm from mathematical standpoint is presented, and finally in this section an optimization procedure of finding the optimal parameters of the RF classifier is given. In Section 3 all experimental results are given, being discussed later in Section 4. Section 5 contains the conclusions over the main results from this study.

2 Dataset and Classifier Description

2.1 Applied Dataset Description

The dataset [21], used within this study, contains records of network connections, established for both typical data exchange and for carrying out IoT-based DDoS attacks from a botnet towards victim machines. It is called Bot_IoT and is publicly available. The latter are Windows 7 workstation, Ubuntu Mobile terminal, an Ubuntu Server with FTP, HTTP, SSH, DNS and E-mail services running on it, and also a Metasploitable. The bots are 4 Kali machines. All victims and bots are connected in internal network with an address space 192.168.100.* through a Local Area Network (LAN) interface. Ubuntu tap machine records all the network traffic for further study.

In Table 1 the relative portion of the normal traffic connections, being totally 9543, is presented.

Table 1. Non-malicious data transfer connections in the Bot-IoT dataset, [21]

Protocol	Relative portion, %
UDP	75.71
TCP	18.34
ARP	4.90
IPv6-ICMP	0.92
ICMP	0.09
IGMP	0.02
RARP	0.01

The typical data, being transferred between the IoT devices, working in normal mode and the server, include information and control signals from smart fridge, a garage door, weather station, adjustable lights, and intelligent thermostat.

The malicious traffic, a result from operating the bots, is summarized in Table 2 as statistics of the attacks, 73360900 in number as a whole.

Table 2. Attacks as a relative portion from all attacks in %, [21]

Reconnaissance	Service scanning (10)		1.99
	OS Fingerprinting (9)		0.49
Denial of Service	DDoS	TCP (4)	26.65
		UDP (5)	25.85
		HTTP (6)	0.03
	DoS	TCP (1)	16.79
		UDP (2)	28.16
		HTTP (3)	0.04
Information Theft	Keylogging (7)		0.0002
	Data Theft (8)		0.0002

There is an excerpt from the whole dataset, used in our experimentation, that includes 2934817 samples as a training set, of which 370 relate to normal traffic, and 733705 samples in a test set with 107 instances of ordinary data transfer. These are all 10-feature vectors with 11th – categorical target variable, which denotes the type of attack as a number (shown in brackets in Table 2). Normal traffic instances are indexed with 0. The features and their distribution is described in Section 2.2.

2.2 Dataset Features and Their Distribution

Each sample from the dataset contains the following 10 features as numeric values: *seq* (sequence) – the sequence identifier of a record, *stddev* (standard deviation) – samples' standard deviation after aggregation, *N_IN_Conn_P_SrcIP* (Number of Incoming Connection per Source IP address) – count of the inwards connections for the source IP address, *min* – minimum value for the time of existence of the registered records, *state_number* – feature state, indexed by a dedicated number, *mean* – mean period, taken by a connection for a particular record to be generated, *N_IN_Conn_P_DstIP* (Number of Incoming Connection per Destination IP address) – the count of inwards connections for the destination IP address, *drate* (destination rate) – destination to source packets rate, *srate* (source rate) – source to destination packets rate, *max* (maximum) – the time taken by the connections, for which the most prolonged records exist.

The initial distribution of the features from the dataset by a type of attack, projected on a 2D space with equidistant position of the 10 components, is given in Fig. 1 a. Most of the samples from the various attacks are being overlapped and not clearly separable in that projection. In order to get the most informative components in front of the least significant one, the Principal Component Analysis (PCA) [22] has been applied over the input set of data. The main stages include the following transformations – from the input data $\boldsymbol{V}$, arranged in a matrix form, separate rows $\boldsymbol{v}_{(n)}$, processed as vectors, are being projected to new vectors $\boldsymbol{s}_{(n)} = (s_1, s_2, \ldots, s_M)_{(n)}$, using q-dimensional vectors of coefficients of proportion $\boldsymbol{p}_{(m)} = (p_1, p_2, \ldots, p_q)_{(m)}$, where $m = 1, 2, \ldots, M$ with the M showing the size of the set, and $n = 1, 2, \ldots, N$ – with the N – the number of principle components. The overall transformation is in the form [22]:

$$s_{m(n)} = \boldsymbol{v}_{(n)}\boldsymbol{p}_{(m)}. \quad (1)$$

The components $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, ..., $\boldsymbol{s}_M$ should have projections of values from the input dataset over them with maximum variance after the transformation. This condition could be met, given [22]:

$$\boldsymbol{p}_{(1)} = \underset{\|\boldsymbol{p}\|=1}{argmax}\left\{\sum_n (s_1)^2_{(n)}\right\} = \underset{\|\boldsymbol{p}\|=1}{\text{argmax}}\left\{\sum_n \left(\boldsymbol{v}_{(n)}.\boldsymbol{p}\right)^2\right\}. \quad (2)$$

Equation (2) includes the total variation of the newly projected n components s_i, denotes now as $\boldsymbol{v}_{(i)}$, over the space, defined by the unit vectors $\boldsymbol{p}$, which yields maximal possible deviation for them.

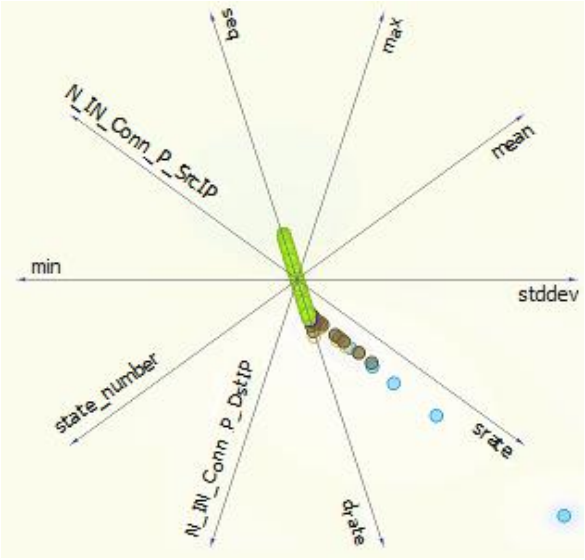
For all subsequent components, it is true [22]:

$$\hat{\boldsymbol{V}}_m = \boldsymbol{V} - \sum_{i=1}^{m-1} \boldsymbol{V}\boldsymbol{p}_{(i)}\boldsymbol{p}_{(i)}^T. \quad (3)$$

The final components of proportion, then, would be [22]:

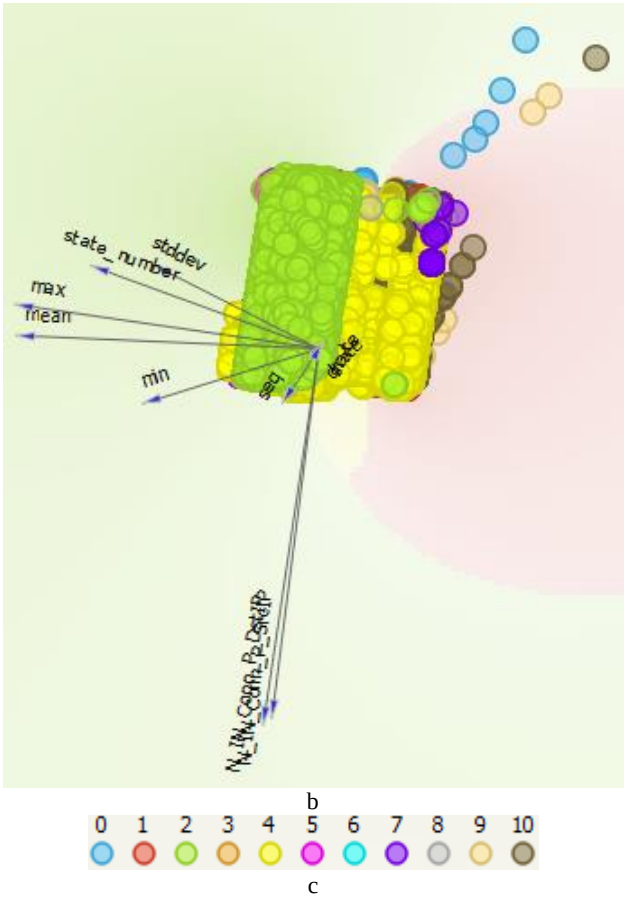
$$\boldsymbol{p}_{(m)} = arg\max\left\{\frac{\boldsymbol{p}^T \boldsymbol{X}_m^T \boldsymbol{X}_m \boldsymbol{p}}{\boldsymbol{p}^T \boldsymbol{p}}\right\}. \quad (4)$$

The result of applying the PCA could be seen in Fig. 1 b.



a

Fig. 1: Training samples, projected on 2D space – a, and after PCA transformation – b, with color legend – c



b

c

Fig. 1 (contn'd): Training samples, projected on 2D space – a, and after PCA transformation – b, with color legend – c

After rotating all initial axes, corresponding to the various features to the designated angles and scaling them with regard to the total variance of the samples, projected over them, it is clearly seen that the *seq* transformed vector is the smallest one and also that the *N_IN_Conn_P_SrcIP* and the *N_IN_Conn_P_DstIP* are very close both in direction and in magnitude (Fig. 1 b). This is the reason to select as a second feature set for testing in this study, apart from the full set of 10 features - *seq, stddev, N_IN_Conn_P_SrcIP, min, state_number, mean, N_IN_Conn_P_DstIP, drate, srate, max*, only 8 of them – the most expressed components – *stddev, min, state_number, mean, N_IN_Conn_P_DstIP, drate, srate, max*. The first 2 Principal Components (PC1 and PC2) by features are given with their magnitudes in Table 3.

Table 3. First two principal components magnitudes by features

Feature	PC1	PC2	Feature	PC1	PC2
stddev	-0.30	0.15	min	-0.32	-0.1
mean	-0.55	0.02	state_number	-0.42	0.15
max	-0.55	0.08	N_IN_DstIP	-0.1	-0.69
seq	-0.07	-0.10	Drate	0.01	0.01
N_IN_SrcIP	-0.09	-0.67	srate	0.006	0.01

The relation of the proportion of the variance to the number of principal components, being preserved for further processing during the classification process, when using just 8 features, is shown in Fig. 2. The cumulative variance is 0.985 and the component variance is just 0.053. More than 98% of the energy, and thus the information, has been preserved in the remaining components of the reduced in dimensionality dataset.

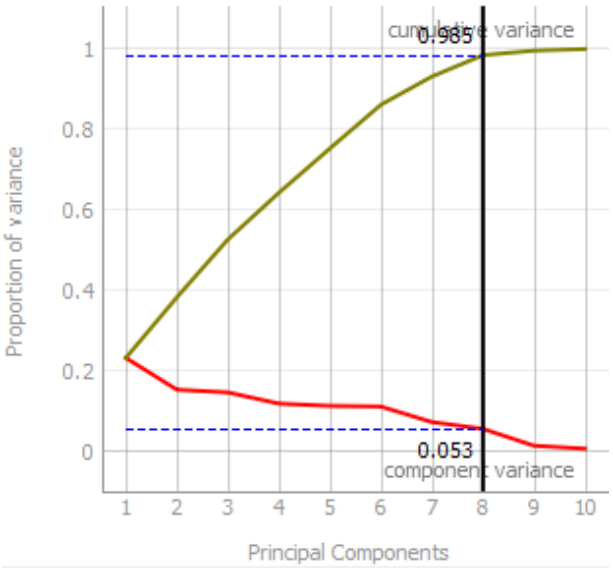


Fig. 2: Proportion of the variance of input data as a function of the number of principal components, with a fixed excerpt at 8 components

2.3 Random Forest Operation Principle

The Random Forest (RF) algorithm [23, 24] is a supervised learning algorithm, which is used for both classification and regression analysis. Each RF has been built by Decision trees (DT) and with the growth of the number of trees, that is more dense becomes the RF, the more robust is the classifier. Subsets of training samples are used to generate the DTs and later predictions are being made by every DT, while the final decision is taken by majority voting. This is an ensemble type of a classifier. Its operation could be represented as 4-step process: starting with random samples from a training set, followed by building a tree for each sample, and making prediction from each tree for getting a decision with a voting among all predictions for every sample, and at the very end the outcomes, which gathered the most votes, are being selected as a final result from the prediction.

Ensemble of classifiers could be defined as $c_1(\boldsymbol{v})$, $c_2(\boldsymbol{v})$, …, $c_k(\boldsymbol{v})$, where the input set of samples v is drawn randomly from the entire training set $C,\boldsymbol{V}$ [24]. $\boldsymbol{V}$ is just the set of training samples, while the complement to C is comprised from the validation and test subsets. The class of a particular attack is denoted with C. Then, a margin function could be proposed, according to [24]:

$$\mathcal{M}(\boldsymbol{V},C) = \langle \mathcal{I}(c_k(\boldsymbol{V}) = C\rangle - max_{j\neq C}\langle \mathcal{I}(c_k(\boldsymbol{V}) = j)\rangle, \quad (5)$$

where $\mathcal{I}$ denotes an indicator function. The increase of the margin corresponds to the overall confidence of classification [24]. The error in the generalization process of the RF algorithm could then be expressed as [24]:

$$\mathcal{E} = P_{V,C}(\mathcal{M}(\boldsymbol{V},C) < 0), \quad (6)$$

where $c_k(\boldsymbol{V}) = c(\boldsymbol{V},\vartheta_k)$. When the number of trees is ever growing, then the following relation holds ever stronger [24]:

$$P_{V,C}(P_\theta(c(\boldsymbol{V},\theta) = C) - max_{j\neq C}P_\theta(c(\boldsymbol{V},\theta) = j) < 0) \quad (7)$$

and this is the reason for a lack of overfitting in the execution of the algorithm.

It has been proven [24], that the error of generalization of the classifier could go up as high as:

$$\mathcal{E}^* \leq \bar{\rho}(1 - s^2)/s^2, \quad (8)$$

where $s = \langle \mathcal{M}(\boldsymbol{V},C)\rangle_{\boldsymbol{V},C}$ is the strength of the classifier and $\bar{\rho}$ – the average magnitude of the correlation coefficient among the different realizations of the raw margin function of the RF implementation. For a multiclass tasks, solved by the RF, the following relation could be used as a base for further estimation of the possible accuracy, as shown in [24]:

$$\mathcal{E}^* \leq \sum_j \sigma^2\{P_\theta(c(\boldsymbol{V},\theta) = C) - P_\theta(c(\boldsymbol{V},\theta) = j)\}/ s_j^2, \quad (9)$$

where $\sigma^2(.)$ is the variance operator, and in the same time $c/s^2 = \bar{\rho}/s^2$. There are certain simplifications of (9) in a 2-class tasks (binary classifier, or also detector), but both cases are of interest to us in this study. In the first case, we classify testing samples into attack and non-attack ones, and in the second case – to non-attack samples and 10 different types of samples, corresponding to 10 different attacks as ordered in Table 2.

Some of the major benefits of using the RF algorithm is that there is no overfitting, the overall good accuracy over big sets of data (higher than that for a single tree), the smaller variance in the results, the lack of necessity to have prior scaling of input variables and the relatively good accuracy, given missing data samples for particular outcomes.

On the other hand, RF are slower to build than separate Decision Trees, and it is harder to interpret the final structure of the forest, compared to DTs. The prediction, being made with RF, is a longer process with regard to the prediction, made by other algorithms.

2.4 Optimization Procedure for the Random Forest Algorithm

There are two tunable parameters in our implementation of the RF algorithm, classifying DDoS attakcs from aggregated network samples (Fig. 3). The first one is the number of trees *t*, which typically is around 10 [25], and in our study it varies between 1 and 11. The other parameter is the minimum number of subsets *s*, which should not be split further during training and testing. This variable we tune from 3 to 7, which range as we shall see in Section 3 causes some actual change of the detection rate of various attacks. All testing has been done with 10 and 8 feature sets, for which purpose the parameter *f* is included in the algorithm chart from below.

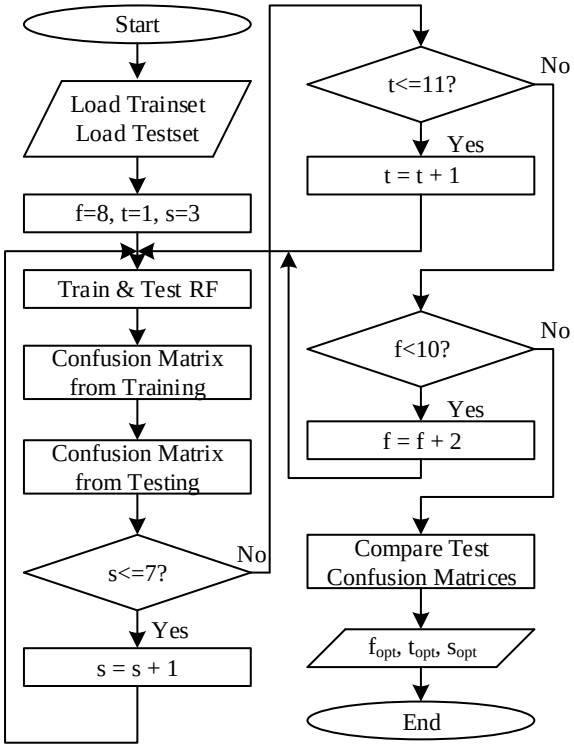


Fig. 3: RF optimization procedure

For each triplet of values <*f*, *t*, *s*> a training, with complete validation over the training set, followed by a testing with the test set is done. Once, the mode is detecting DDoS attacks vs. normal traffic discovery and secondly – classifying all present attacks in the dataset with clear distinction of the non-attack samples. The confusion matrices from validation and test phase are being calculated and compared later for finding the optimal set of parameters for both modes of operation of the RF algorithm – f_{opt}, t_{opt}, and s_{opt}.

The following parameters are used for evaluating the efficiency of the classifiers in the two modes of operation – Area Under the Curve (Receiver Operating Characteristic) – *AUC*, Classification Accuracy (*CA*), *F1*-measure, *Precision*, *Recall*, *Log-loss*, *Specificity* [26]. Training and testing time over the train set and test set, respectively, for each realization of the RF algorithm, and the confusion matrix for every case are the other set of parameters, found for comparative analysis.

3 Experimental Results

The hardware setup, used during testing is described in Table 4. The software for implementation of the RF algorithm is Orange v. 3.28, running over 64-bit MS Windows Professional 10. Orange is freely redistributable software. It allows for visual programming and ease of use, which are the main factors for selecting it as a working environment. One of its drawbacks is the inability to introduce changes in particular classification function without recompiling the whole application.

Table 4. Hardware test platform

CPU	Frequency	CPU cache			RAM	HDD
		L1	L2	L3		
Xeon E5-1620	3.5 GHz (4 cores)	256 kB	1 MB	10 MB	64 GB	2 TB 7200 rpm

Running the optimization procedure, proposed in Section 2.4 (Fig. 3), led to the results, presented in Fig. 4 as per the number of trees tried, when performing the detection process (binary classification).

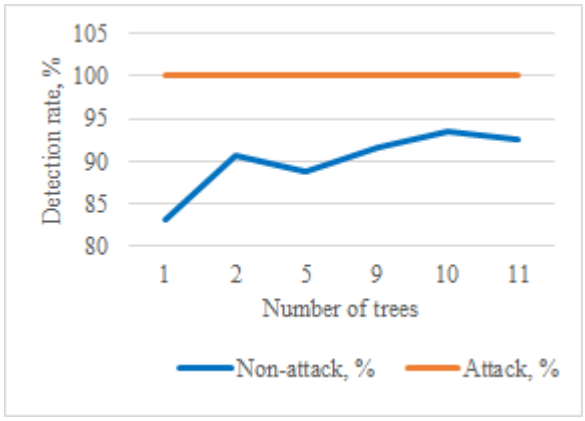


Fig. 4: Detection rate of DDoS attacks of the number of trees

The detection rate of the attacks remains constant and almost equal to 100% in virtually all cases, when the number of trees is varied between 1 and 11. In the same time, there is a peak at 10 trees, reaching detection rate of around 94% for the non-attack samples. Because of that, topt = 10 for all further experiments.

The dependency of the detection rate as a function of the minimal number of subsets to split (Fig. 5) offers a bit different picture to that of the total number of trees as independent parameter. Again, the detection rate is constant and almost equal to 100% in virtually all cases, but the success into detecting the non-attack samples goes into a saturation level for the interval between 4 and 6 subsets – close to 94%. Prior and after that interval, there are drops in the detection rate, going down to almost 90%. A selection has been made in the center of that saturation interval, that is s_{opt} = 5, which is preferred for further testing in the rest of the experiments.

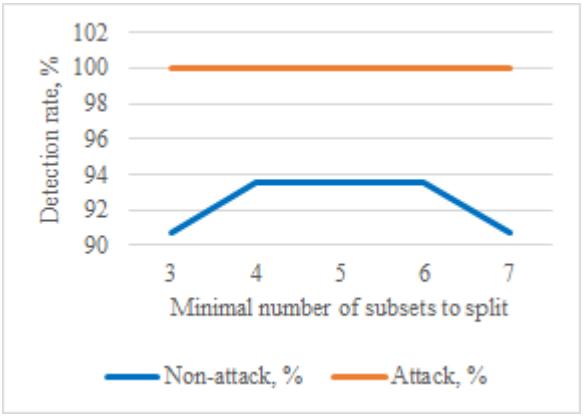


Fig. 5: Detection rate of DDoS attacks of the number of subsets

Detecting DDoS attacks regardless of their type and discriminating them from the non-attack samples, when applying all 10 features, leads to the evaluating parameters, given in Table 5. The Class (Cl.) labels of 0 and 1 correspond to normal and attack malicious traffic, respectively. Average performance has been also calculated for both classes, denoted with Av. All parameters are found from validation over the complete train set (Train) and from testing over the complete test set (Test).

Table 5: DDoS detection efficiency using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Train	0	1.0000	0.9999	0.9973	0.9946	1.0000	1.5494	0.9999
	1	1.0000	0.9999	0.9999	1.0000	0.9999	1.5494	1.0000
	Av.	1.0000	0.9999	0.9999	0.9999	0.9999	1.5494	0.9999
Test	0	0.9999	0.9999	0.9615	0.9901	0.9346	3.5837	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	3.5837	0.9346
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	3.5837	0.9346

Evaluating parameters with their values when applying the RF detector with only 8 features, as described in Section 2.2, are given in Table 6.

Table 6: DDoS detection efficiency using 8 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Train	0	0.9999	0.9999	0.9850	0.9918	0.9784	1.6839	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	1.6839	0.9784
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	1.6839	0.9784
Test	0	0.9999	0.9999	0.9717	0.9810	0.9626	3.1653	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	3.1653	0.9626
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	3.1653	0.9626

The confusion matrices from detecting attacks over the test set, using 10 and 8 features are graphically presented in Fig. 6 a and b, respectively.

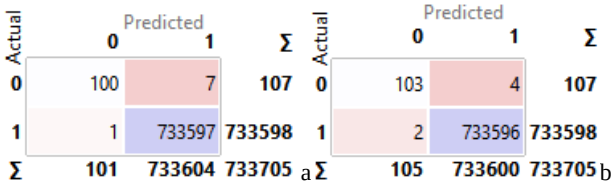


Fig. 6: Confusion matrices from the test set of detecting DDoS attacks at a – 10 features, b – 8 features

After carrying out multiclass discrimination, that is full classification over 10 types of attacks (indexes 1-10 as described in Table 2) and non-attack instances (index 0), the result when validating over the train set is visible in Table 7. That experiment has been done for 10 features. The class category, as in the detector case, is shortly denoted here with the Cl. abbreviation as well (for 0-10).

Table 7. DDoS classification efficiency from full validation using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Training	0	0.9999	0.9999	0.9959	0.9919	1.0000	1.6509	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	2.9639	0.9999
	2	0.9999	0.9999	0.9999	0.9999	0.9999	1.0465	0.9999
	3	0.9999	0.9999	0.9996	0.9992	1.0000	1.1980	0.9999
	4	0.9999	0.9999	0.9999	0.9999	0.9999	2.9664	0.9999
	5	0.9999	0.9999	0.9999	0.9999	0.9999	0.5259	0.9999
	6	1.0000	0.9999	0.9994	1.0000	0.9987	0.8330	1.0000
	7	0.9999	0.9999	0.9915	1.0000	0.9831	0.4201	1.0000
	8	1.0000	1.0000	1.0000	1.0000	1.0000	0.1726	1.0000
	9	0.9999	0.9998	0.9821	0.9820	0.9822	78.406	0.9999
	10	0.9999	0.9998	0.9956	0.9957	0.9956	78.389	0.9999
	Av.	0.9999	0.9998	0.9998	0.9998	0.9998	84.585	0.9999

The next experiment is realized with classification of the test samples (from the test set) of the same RF model, using 10 features, after its training over the larger train set, and the resulting values of the evaluation parameters are presented in Table 8.

Also, in both Table 7 and Table 8, the average values of all parameters – from *AUC* to *Specificity* are found and shown in the last row of the tables, denoted with the abbreviation of Av.

Table 8. DDoS classification efficiency over unknown samples using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Testing	0	0.9999	0.9999	0.9423	0.9703	0.9159	4.8651	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	15.715	0.9999
	2	0.9999	0.9999	0.9999	0.9999	0.9999	2.8318	0.9999
	3	0.9999	0.9999	0.9901	0.9836	0.9967	2.5862	0.9999
	4	0.9999	0.9999	0.9999	0.9999	0.9999	20.388	0.9999
	5	0.9999	0.9999	0.9999	0.9999	0.9999	1.4053	0.9999
	6	0.9975	0.9999	0.9950	1.0000	0.9901	6.5501	1.0000
	7	0.9999	0.9999	0.9630	1.0000	0.9286	0.6696	1.0000
	8	0.9971	0.9992	0.9230	0.9339	0.9125	295.77	0.9997
	9	0.9997	0.9992	0.9808	0.9779	0.9838	297.33	0.9995
	10	0.9999	0.9998	0.9956	0.9957	0.9956	78.389	0.9999
	Av.	0.9994	0.9992	0.9992	0.9992	0.9992	329.93	0.9999

In Table 9 the results from validation with the full train set are given.

Table 9. DDoS classification efficiency from full validation using 8 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Training	0	0.9999	0.9999	0.9893	0.9788	1.0000	1.8002	0.9999
	1	0.9999	0.9992	0.9977	0.9971	0.9982	152.82	0.9994
	2	0.9999	0.9999	0.9999	0.9999	0.9999	1.2414	0.9999
	3	0.9999	0.9999	0.9992	1.0000	0.9983	1.9361	1.0000
	4	0.9999	0.9990	0.9981	0.9981	0.9982	270.42	0.9993
	5	0.9999	0.9999	0.9999	0.9999	0.9999	0.6737	0.9999
	6	0.9999	0.9999	0.9994	1.0000	0.9987	1.7368	1.0000
	7	0.9999	0.9999	0.9508	0.9206	0.9831	0.9243	0.9999
	8	0.9999	1.0000	1.0000	1.0000	1.0000	0.2020	1.0000
	9	0.9974	0.9959	0.4248	0.6688	0.3112	792.74	0.9992
	10	0.9994	0.9959	0.9037	0.8553	0.9579	778.18	0.9967
	Av.	0.9979	0.9950	0.9944	0.9945	0.9950	1009.0	0.9996

In Table 10 are presented the values of evaluating parameters when classifying the test set with 8 features.

Table 10. DDoS classification efficiency over unknown samples using 8 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Testing	0	0.9999	0.9999	0.9528	0.9619	0.9439	4.8993	0.9999
	1	0.9998	0.9988	0.9966	0.9962	0.9969	510.33	0.9992
	2	0.9999	0.9999	0.9999	0.9999	0.9999	2.7224	0.9999
	3	0.9999	0.9999	0.9901	0.9868	0.9934	3.8869	0.9999
	4	0.9998	0.9986	0.9975	0.9973	0.9976	628.37	0.9990
	5	0.9999	0.9999	0.9999	0.9999	0.9999	1.9314	0.9999
	6	0.9951	0.9999	0.9876	0.9950	0.9803	12.708	0.9999
	7	0.9643	0.9999	0.9231	1.0000	0.8571	5.8171	1.0000
	8	0.9756	0.9952	0.3475	0.5373	0.2568	1614.0	0.9989
	9	0.9987	0.9953	0.8873	0.8397	0.9405	1603.5	0.9964
	10	0.9994	0.9959	0.9037	0.8553	0.9579	778.18	0.9967
	Av.	0.9938	0.9939	0.9933	0.9932	0.9939	469.67	0.9995

Execution (processing) times are being measured both during the train and the test phases for both sets of features – 10 and 8 in number (Table 11). Testing is split in 2 parts. The first one is the validation over the complete training set after the training itself, and the second one is the

Table 11. Proposed RF classifiers processing times

RF mode	Features number	Training time, s	Validation time, s	Testing time, s
Detection	8	179.94	15.00	6.21
	10	335.88	31.24	7.87
Classification	8	311.35	54.94	13.63
	10	399.77	55.14	12.69

The Confusion Matrices from conducting the testing phase with just unknown to the RF samples are visualized in Fig. 7 for 10 features and in Fig. 8 – for 8 features.

The indexes of attacks are over the horizontal and vertical first series and inside the matrices are the absolute number of classified instances of particular type. Below and to the right of each matrix are the total number of attacks by type being predicted (below) and the actual one (on the right side).

Actual \ Predicted	0	1	2	3	4	5	6	7	9	10	Σ
0	98	0	2	0	0	0	0	0	1	6	107
1	1	123180	0	1	3	0	0	0	0	0	123185
2	0	4	206619	2	0	1	0	0	0	0	206626
3	1	0	0	300	0	0	0	0	0	0	301
4	0	4	1	0	195147	0	0	0	0	0	195152
5	0	0	1	0	0	189952	0	0	0	1	189954
6	0	0	0	0	1	0	201	0	1	0	203
7	0	0	0	0	0	0	0	13	0	1	14
9	0	1	0	0	0	0	0	0	3304	316	3621
10	1	0	0	2	0	0	0	0	232	14307	14542
Σ	101	123189	206623	305	195151	189953	201	13	3538	14631	733705

Fig. 7: Confusion matrices from the test set of classifying DDoS attacks using 10 features

Comparison with comments between the resulting matrices could be found in Section 4, where a discussion has been made. The total number of test instances is in the bottom-right corner.

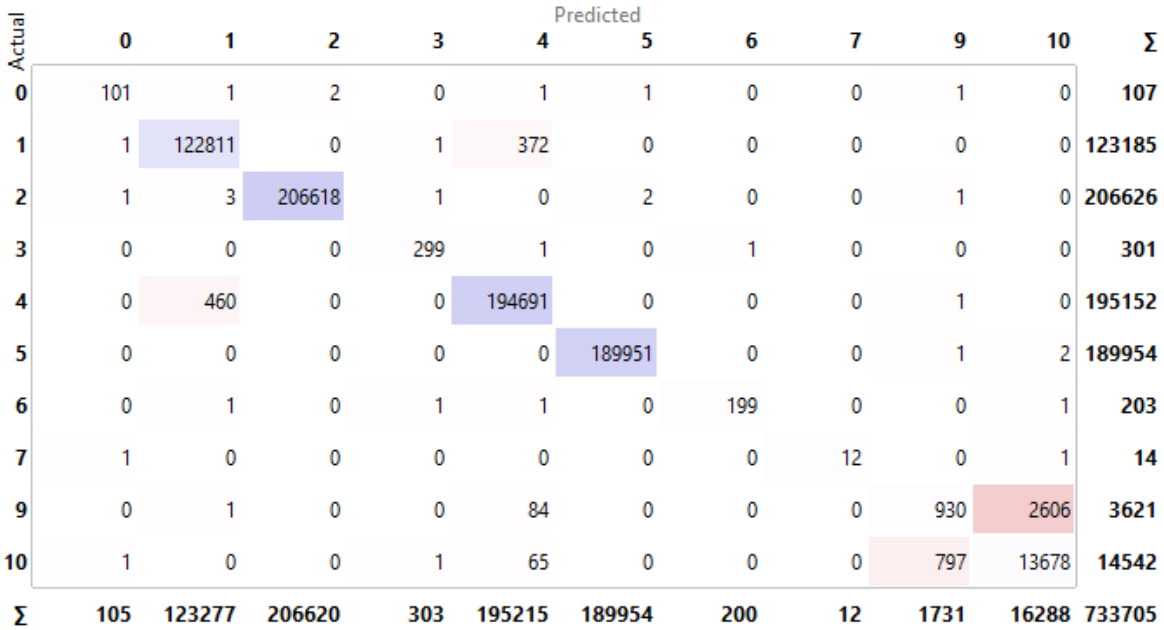


Fig. 8: Confusion matrices from the test set of classifying DDoS attacks using 8 features

4 Discussion

Below is given the relation between observations from Section 3 regarding the reduced combined set of 8 features and the performance of the classifier. Then, it is compared to the performance when using the full set of 10 features.

The first observation of the behavior of the RF DDoS attack detector is related to the slight difference in performance between discriminating the non-attack samples and attacks ones with a difference in the *F1*-measure of around 0.0026 during validation (Table 5). During testing this difference gets higher – around 0.0384. With the exception of the slight variation of *AUC* and some other parameters, the first of which is 0.0001 higher during training, compared to testing, most of the other parameters are relatively stable between these two phases. It means that the RF detector is stable and is slightly more inaccurate for non-attack samples, than for attack ones, when employing the full set of 10 features. *AUC* and *CA* at 8 features are almost identical to those at 10 features (Table 6 vs. Table 5). *F1*-score falls a bit with 0.0123 during training, but it's catching up with 0.0102 when testing with unknown samples, compared to the value of detecting at 10 features. Most of the other parameters are relatively equal between the two cases of 8 and 10 features. This leads to the conclusion that in its optimal configuration, derived from optimization according Fig. 3, the use of 8 features for DDoS detection by the RF binary classifier is efficient enough and could be used with the benefit of processing smaller amount of data – by 2 components from a sample. This property is supported also by the results, contained in the Confusion Matrices on Fig. 6. Only 7 out of 107 non-attack samples are misclassified, using 10 features, and there are just 4 out of 107 misclassifications of normal traffic instances, using 8 features, which is even lower error rate. Only 1 instance from the attack samples more is being wrongly classified on the background of 733598 present malicious records, given 8 features, with a single misclassification at 10 features, which in both cases is really negligible. Training the RF with 8 features as a detector is 1.87 times faster than the model with 10 features (Table 11). Validation process takes almost twice less time, using 8-featured vector samples, than 10-featured ones. These relations, connected to the processing times of the tested RF DDoS detectors, along with the detection accuracy, reveal that the 8-featured version is fully applicable as an implementation for real-world monitoring of the attack types at hand and could be a substitute of the 10-feature version, which is also fully admissible for the same task.

Classification accuracy of the multiclass RF model is very close among all types of attacks, both during training and testing (Table 7 and Table 8). For the Data Theft (8) and OS Fingerprinting (9) attacks the discovery rate is about 0.07% less to all the others with indexes from 1 to 7, and for the Service Scanning (10) – it is 0.01% less. These differences could be explained to some extent with the lower relative portion of samples from these attacks, being much less intensive than the DoS and DDoS floods. *F1*-measure and the rest parameters vary a bit more during the testing phase from attack

to attack, with a value for the first of 0.0007. In general, the classification ability of the RF model at 10 features remain robust regardless of the type of attack to spot. Classification accuracy, when using 8 features (Tables 9 and 10), and especially during the test phase (Table 10) varies more – for the DoS TCP (1) with 0.11% less, DDoS TCP (4) – with 0.13% less, Data Theft (8) – with 0.47% less, OS Fingerprinting – with 0.46% less, and for the Service Scanning – with 0.40% less than the rest of the attacks, for which *CA* is 0.9999. *F1*-measure in the same time deviates with a variance of 0.0363. It definitely shows that the RF multi-attack classifier, using 8 features, is not that robust as the 10-featured one. In confirmation to these generalized numbers, a more detailed look into the recognition rate of the various attacks, visible from the Confusion Matrices (Fig. 7 and Fig. 8), the following tendencies could be noted:

- At 10 features – 91.6% of the normal traffic samples (0) are correctly classified with 5.6% erroneously marked as Service Scanning (10) attack as the major group of misclassification; 100 % of DoS TCP (1) and UDP (2), DDoS TCP (4) and UDP (5) samples are correctly recognized; 99.7% of the DoS HTTP (3) are correctly recognized with just 0.3% misclassifications as normal traffic (0) samples; DDoS HTTP (6) attacks are found in the 99.0% of the cases with 0.5% misclassifications in equal proportion between DDoS TCP (4) and OS Fingerprinting (9) attacks; 92.9% is the recognition rate of the Keylogging (7) with 7.1% of the samples categorized as Service Scanning (10); 91,2% of the samples, when performing OS Fingerprinting (9), are correctly discovered with 8.7% of them wrongly marked as Service Scanning (10) activity; at last, 98.4% of the Service Scanning (10) samples are correctly found, but 1.6% are labeled as OS Fingerprinting (9);
- At 8 features – 94.4% correctly found non-attack samples (0) – the larger group of misclassifications, equal to 1.9% of the normal samples is recognized as DoS UDP (2) records; 99.7% of the DoS TCP (1) samples are recognized with 0.3% - marked as DDoS TCP (4) instances; 100% of the DoS UDP (2) and the DDoS UDP (5) samples are discovered; 99.3% of the DoS HTTP (3) with 0.3% misclassifications as DDoS TCP (4) and DDoS HTTP (6) attacks; 99.8% - correct DDoS TCP (4) attack vectors vs. 0.2% recognized wrongly as DoS TCP (1); almost the same result of 98% truly found DDoS HTTP (6) flood related samples against 0.5% incorrectly found ones, equally spread among DoS TCP (1), DoS HTTP (3), DDoS TCP (4) and Service Scanning (10) attacks; 85.7% correct Keylogging activities with 7.1% mistakenly fixed as non-attacks (0) and Service Scanning (10) samples; only 25.7% of the OS Fingerprinting samples are recognized, which is the only type of attack with that significantly low recognition rate – 72.0% of the instances of this type are listed to be Service scanning (10) samples – further examination of the feature distribution when using the full set of 10 components against the 8-featured samples would reveal which exactly of the 2 missing elements are contributing the most to have that reduction of 65.5% drop in the recognition rate for this particular attack; 94.1% of the Service Scanning (10) samples are correctly found with 5.5% as the largest group of erroneous samples being marked as OS Fingerprinting (9).

Multiclass training of the RF at 10 features takes 1.28 times longer than the 8-featured implementation, but the validation process and testing with unknown samples (test set) do not produce any noticeable time difference between the two models (Table 11). Pursuing the most accurate result in recognizing the precise type of attack, it would be preferable to use the 10-featured RF model. On the other hand, only the Keylogging (7) (by 7.2% decrease) and the OS Fingerprinting (9) (by 65.5% decrease) are being significantly misclassified by the 8-featured RF model. In cases, where these type of activities are not expected to emerge, and given the smaller portions of data to handle at 8 features, it would be preferable to use that particular model.

Comparison with another RF model, developed to detect DDoS malicious actions from network traffic aggregated samples [27] and the proposed in this study 8- and 10-component RF detectors of DDoS attacks, is given in Table 12. For all 4 evaluating parameters – *CA*, *Precision*, *Recall* and *F1-score*, there has been an increase in the registered values for the newly proposed detectors, which proves the applicability of the optimization procedure and feature selection methods, suggested here.

Similar, although smaller, are the ratios between the evaluating parameters for the RF model from [27] and the two implementations proposed here, when processing multiclass samples (Table 13) – another confirmation of the correctness of the undertaken approach.

Table 12. Comparison between two RF models in detection mode

Parameter	RF, [27]	Proposed RF - 10 features	Proposed RF - 8 features
CA	0.9532	0.9999	0.9999

Precision	0.9580	0.9999	0.9999
Recall	0.9532	0.9999	0.9999
F1-score	0.9481	0.9999	0.9999

Comparing the 10-featured version of a detector from this study and the already developed one [27] shows increase of 4.67% for the *CA*, 4.19% - for the *Precision*, 4.67% - for the *Recall*, and 5.18% - for the *F1*-score. When classifying multiple attacks, these increments are: 2.26%, 1.68%, 2.26%, and 3.35%, respectively.

Table 13. Comparison between two RF models in classification mode

Parameter	RF, [27]	Proposed RF - 10 features	Proposed RF - 8 features
CA	0.9766	0.9992	0.9939
Precision	0.9824	0.9992	0.9932
Recall	0.9766	0.9992	0.9939
F1-score	0.9657	0.9992	0.9933

Similar comparison could be made with earlier works in the field [28-32].

5 Conclusion

In this paper two new Random Forest models are proposed, based on 8- and 10-component features from aggregated network connection records, capable to detect and classify 10 types of DDoS attacks apart from normal traffic. There is no significant difference in terms of detection efficiency between the 8- and 10-featured implementations of detectors. The first is preferred for application due to the smaller amount of data to process and the smaller training and testing time. Classification accuracy rises significantly for Service Scanning and OS Fingerprinting attacks when using 10 features by the multiclass RF classifier. That would be the preferable classifier to use for the full spectrum of attacks under consideration. Due to the smaller processing times at 8 features and the smaller amounts of handled data, there could be practical cases when this classifier would be also preferable to use. Further work is needed in order to find the optimal set of features, representing the OS Fingerprinting and Keylogging attacks, where less than 10 components would lead to comparable, high enough classification rate for them, as it is for the rest 8 types of investigated attacks.

Incorporation of the proposed RF classifier in multicomponent classifiers of DDoS network attacks is one possible direction for a future work. Extending the feature set to more than 10 elements could allow for a broader list of attacks to be discovered by either the standalone RF classifier or by a cascade of classifiers. Wider investigation of dependencies among raw traffic parameters is needed in order to define this set of new features. Future work would be needed in this case in order to optimize the structure of randomly generated decision trees to retain the processing time as low as possible and allow for the new classifiers to stay applicable in the practice.

References:

[1] Dong, S., Abbas, K., Jain, R. A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments. *IEEE Access*, Vol. 7, 2019, pp. 80813-80828.

[2] Bindra, N., Sood, M. Detecting DDoS Attacks using Machine Learning Techniques and Contemporary Intrusion Detection Dataset. *Automatic Control and Computer Sciences*, Vol. 53, No. 5, 2019, pp. 419-428.

[3] Idhammad, M., Afdel, K., Belouch, M. Detection System of HTTP DDoS Attacks in a Cloud Environment based on Information Theoretic Entropy and Random Forest. *Security and Communication Networks*, Vol. 2018, Article ID 1263123, 2018.

[4] Chen, L., Zhang, Y., Zhao, Q., Geng, G., Yan, Z. Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark. *Procedia Computer Science*, Vol. 134, 2018, pp. 310-315.

[5] Lu, L., Feng, Y., Sakurai, K. C&C Session Detection using Random Forest. *In Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, 2017, pp. 1-6.

[6] Farukee, M. B., Shabit, M. Z., Haque, M. R., Sattar, A. S. DDoS Attack Detection in IoT Networks Using Deep Learning Models Combined with Random Forest as Feature Selector. *In International Conference on Advances in Cyber Security*, Springer, Singapore, 2020, pp. 118-134.

[7] Pande, S., Khamparia, A., Gupta, D., Thanh, D. N. DDoS Detection using Machine Learning Technique. *In Recent Studies on Computational Intelligence*, Springer, Singapore, 2021, pp. 59-68.

[8] Lopez, A. D., Mohan, A. P., Nair, S. Network Traffic Behavioral Analytics for Detection of DDoS Attacks. *SMU Data Science Review*, Vol. 2, Issue 1, Art. No. 14, 2019.

[9] Hosseini, S., Azizi, M. The Hybrid Technique for DDoS Detection with Supervised Learning Algorithms. *Computer Networks*, Vol. 158, 2019, pp. 35-45.

[10] Sharafaldin, I., Lashkari, A. H., Hakak, S., Ghorbani, A. A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. *In 2019 International Carnahan Conference on Security Technology (ICCST)*, IEEE, 2019, pp. 1-8.

[11] Nurwarsito, H., Nadhif, M. F. DDoS Attack Early Detection and Mitigation System on SDN using Random Forest Algorithm and Ryu Framework. *In 2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, IEEE, 2021, pp. 178-183.

[12] Bakker, J. N., Ng, B., Seah, W. K. Can Machine Learning Techniques be Effectively used in Real Networks against DDoS Attacks?. *In 2018 27th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, 2018, pp. 1-6.

[13] Alkasassbeh, M., Al-Naymat, G., Hassanat, A., Almseidin, M. Detecting Distributed Denial of Service Attacks using Data Mining Techniques. *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 1, 2016, pp. 436-445.

[14] Kumar, P., Kumar, R., Gupta, G. P., Tripathi, R. A Distributed Framework for Detecting DDoS Attacks in Smart Contract- based Blockchain- IoT Systems by Leveraging Fog computing. *Transactions on Emerging Telecommunications Technologies*, Vol. 32, No. 6, 2021, e4112.

[15] Ustebay, S., Turgut, Z., Aydin, M. A. Intrusion Detection System with Recursive Feature Elimination by using Random Forest and Deep Learning Classifier. *In 2018 international congress on big data, deep learning and fighting cyber terrorism (IBIGDELFT)*, IEEE, 2018, pp. 71-76.

[16] Prasad, G., Sandhia, G. K., Sharma, A. Mobile Application based Detection of DDoS Attack with Enhanced Random Forest Algorithm. *Turkish Journal of Physiotherapy and Rehabilitation*, Vol. 32, No. 3, 2021, pp. 3263-3271.

[17] Nandi, S., Phadikar, S., Majumder, K. Detection of DDoS Attack and Classification using a Hybrid Approach. *In 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, IEEE, 2020, pp. 41-47.

[18] Kerim, B. Securing IoT Network against DDoS Attacks using Multi-agent IDS. In *Journal of Physics: Conference Series*, IOP Publishing, Vol. 1898, No. 1, p. 012033, 2021.

[19] Gaur, V., Kumar, R. Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices. *Arabian Journal for Science and Engineering*, 2021, pp. 1-22.

[20] Osorio, J. S. M., Tejada, J. A. V., Vega, J. F. B. Detection of DoS/DDoS Attacks: the UBM and GMM Approach. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, IEEE, 2021, pp. 866-871.

[21] Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B., Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT dataset. *Future Generation Computer Systems*, Vol. 100, November 2019, pp. 779-796.

[22] Jolliffe, I. T. *Principal Component Analysis. Springer Series in Statistics*. New York, Springer-Verlag, 2002.

[23] Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests. In Ensemble machine learning (pp. 157-175). Springer, Boston, MA.

[24] Breiman, L. *Random Forests*. Machine learning, Vol. 45, No. 1, 2001, pp. 5-32.

[25] Random Forest, Orange Visual Programming, https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/randomforest.html, last accessed on August 12th, 2021.

[26] Test and Score, Widgets, https://orangedatamining.com/widget-catalog/evaluate/testandscore/, last accessed on August 12th, 2021.

[27] Guerra-Manzanares, A., Medina-Galindo, J., Bahsi, H., Nõmm, S. MedBIoT: Generation of an IoT Botnet Dataset in a Medium-sized IoT Network. In *International Conference on Information Systems Security and Privacy ICISSP 2020*, 2020, pp. 207-218.

[28] Robinson, R. R., & Thomas, C. Ranking of machine learning algorithms based on the performance in classifying DDoS attacks. In *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, IEEE, 2015, pp. 185-190.

[29] Bandara, K. R. W. V., Abeysinghe, T., Hijaz, A., Darshana, D. G. T., Aneez, H., Kaluarachchi, S. J., Sulochana, K. V. D. L., DhishanDhammearatchi, M. Preventing DDOS

attack using data mining algorithms. *International Journal of Scientific and Research Publications*, Vol. 6, No. 10, 2016, pp. 390-400.

[30] Revathi, S., Malathi, A. Detecting Denial of Service Attack Using Principal Component Analysis with Random Forest Classifier. *Int. J. Comput. Sci. Eng. Technol (IJCSET)*, 5, 2014, pp. 248-252.

[31] Boroujerdi, A. S., Ayat, S. A robust ensemble of neuro-fuzzy classifiers for DDoS attack detection. In *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, IEEE, 2013, pp. 484-487.

[32] Kirubavathi Venkatesh, G., Anitha Nadarajan, R. HTTP botnet detection using adaptive learning rate multilayer feed-forward neural network. In *IFIP International Workshop on Information Security Theory and Practice*, Springer, Berlin, Heidelberg, 2012, pp. 38-48.