INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING
DOI: 10.46300/9106.2022.16.93 Volume 16, 2022

IoT-based Network Attacks Discovery with Combined Classifiers

Vanya Ivanova, Tasho Tashev, Ivo Draganov
PhD School, French Faculty of Electrical Engineering
Technical University of Sofia
8 Kliment Ohridski Blvd., 1756 Sofia
Bulgaria

Received: July 21, 2021. Revised: January 15, 2022. Accepted: February 2, 2022.
Published: February 28, 2022.

Abstract— In this paper following the recent trends in IoT-based network attacks discovery and advancing further our previous research, in which we optimize and test single neural network, support vector machine and random forest classifiers for both the detection and recognition of multiple DDoS attacks, we propose results from newly developed combined classifiers. The first of them employs only a neural network and a random forest classifier, while the second use additionally a support vector machine. Both are implemented in two modifications – as detectors of malicious vs. normal traffic, and as classifiers of 10 types of attacks vs. non-attack samples. High classification accuracy is being obtained over the popular Bot-IoT dataset and it prove higher than that of the single classifiers. At the same time, it is also higher than other solutions, proposed in the practice.

Keywords— DDoS, IoT, network attack, combined classifier, neural network, random forest, support vector machine

I. Introduction

IoT-based botnet attacks affect the normal operation of numerous systems over public and private networks on a large scale [1]. Their negative effect renders unusable or highly limits the access to services, offered either from single machines or in distributed environment, including cloud infrastructure. One of the steps towards mitigating their influence is the on-time discovery of malicious network traffic, flowing between targeted and attacking machines. Machine learning offers as perspective mean into solving this task combined classifiers.

In [2] Koay et al. propose entropy-based features within a classifier, consisting of Recurrent Neural Network (RNN), Multi-Layer Perceptron (MLP) Neural Network (NN) and Alternating Decision Tree (ADT). True Positive Rate of the attacks detection reaches 94.74%, compared to 90.04% for non entropy-based methods. Das et al. [3] undertake the opposite approach of trying to identify reduced set of features with a number of ensemble models, each one aimed at different kind of attack, in order to increase detection accuracy. It is being reported as 99.1% when fusing the operation of MLP, Sequential Minimal Optimization (SMO) for Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), and Decision Trees (DT) through the C4.5 (J48) algorithm. Detection rate of the single classifiers, such as k-NN and J48, does not exceed 97.8%. As Musumeci et al. [4] demonstrate, detection of DDoS based attacks could be implemented through the P4 language. They use Random Forest (RF), k-NN and SVM in order to process window features, derived from the input of a P4 switch and then place a decision if the outgoing traffic should be allowed. RF has 98.5% detection accuracy alone. Mahfouz et. al [5], following their earlier research from [3], propose the use only of MLP, k-NN and J48 for the same purpose. Botnet attacks are being discovered with accuracy of 96.20% by J48, 96.50% - by k-NN (IBK implementation), 93.50% - by the MLP, and 98.87% - by ensemble of these classifiers. Normal traffic and DDoS attacks of general type are detected with similar rates, but Brute Force and Infiltration activities are spotted with accuracy in the range of 91.60-96.70% and 80.57-88.69% among the tested classifiers, where further work could be undertaken to increase the performance.

Algelal et al. [6], in a recent study, suggest the use of boosting, bagging and RF altogether with the base classifiers of JRip, Naïve Bayes (NB), and REPTree for detecting botnet activities. The detection accuracy among the three base classifiers, when using boosting, varies between 85.48% and 99.84%, the same range for bagging and the RF alone leads to a result of 95.11%. Khraisat et al. [7] take a bit different approach by constructing 3-stage ensemble for classification of IoT-based attacks. In the 1st stage they use the C5 algorithm, for which the F1-measure into discovering DDoS attacks is

0.998. Reconnaissance activities yield 0.327. The 2nd stage incorporates SVM leading to F1 for Intrusion samples of 0.935, and for normal traffic – 0.920. The final, 3rd stage, combines the C5 and SVM together. Reconnaissance F1 measure is 0.353. The overall accuracy for stage 1 is 93.30%, for 2 – 92.50%, and for 3 – 99.97%. Rajagopal et al. [8] implement a stacking ensemble, consisting of RF, Logistic Regression (LR) and k-NN with which they achieve accuracy of binary classification over multiple network attacks of 0.94. Classifying 10 types of attacks the ensemble achieves Precision between 41.6% and 99.42% among them with Recall, varying between 10.79% and 98.32%. Iwendi et al. [9] apply correlation-based feature selection prior classifying network attacks, part of which Denial of Service (DoS) attempts, with bagging and Adaboost classifiers. Detection rate varies between 98.60% and 99.90% between two datasets used. Another research, described by Jain et al. [10], employs the NB, SVM, k-NN, and RF classifiers for discovering DDoS attacks and simultaneously they are tried as ensemble. The combined classifier yielded a F score of 0.9962, while the NB as least performing discriminator achieved 0.9942 for the same parameter. Zhou et al. [11] use C4.5, RF and Forest by Penalizing Attributes (ForestPA) in ensemble for attacks recognition. Applying the correlation feature selection method and reducing the initial set from 41 to only 10 features, they are able to achieve detection rate of 0.998 for 4 types of attacks and normal traffic samples.

Following the recent trends in IoT-based network attacks discovery, some of which described above, and advancing further our previous research [12, 13, 14, 15], in which we optimize and test single NN, SVM and RF classifiers for both the detection and recognition tasks, in this paper we propose results from newly proposed combined classifiers. The first of them employs only NN and RF classifiers, while the second – NN, RF, and SVM. Both are implemented in two modifications – as detectors of malicious vs. normal traffic, and as classifiers of 10 types of attacks vs. non-attack samples. High classification accuracy is being obtained over the popular Bot-IoT dataset [16] and it prove higher than that of the single classifiers. In the same time, it is also higher than other solutions, proposed by other authors.

In Section 2, the description of the binary and multi-attack classifiers is given, followed by the experimental results in Section 3. They are discussed in Section 4 and conclusions follow in Section 5.

II. Combined Classifiers Description

A. Combined binary classifier

The precise form of the likelihood function of the combined binary classifier (detector), proposed in this section, could be derived, as shown below, following the general approach, given in [17].

The output of the single detectors NN, RF and SVM are o1 ∈ {0, 1}, o2 ∈ {0, 1} and o3 ∈ {0, 1}, respectively. The value of 0 denotes sample from normal traffic and 1 – an attack. The output from the Logistic Regression module is O ∈ {0, 1}. The probability to have an attack, given the decision of the combined classifier is p=P(O=1). According to the structure of the detector from Fig. 1, the log-odds is:

$$l = ln\frac{p}{1-p} = k_0 + k_1o_1 + k_2o_2 + k_3o_3, \quad (1)$$

where k_0, k_1, k_2, and k_3 are the parameters of the detection model. Raising the left and right side of (1) over e leads to:

$$\frac{p}{1-p} = e^{k_0+k_1o_1+k_2o_2+k_3o_3}, \quad (2)$$

which in turn gives:

$$p = \frac{e^{k_0+k_1o_1+k_2o_2+k_3o_3}}{e^{k_0+k_1o_1+k_2o_2+k_3o_3}+1} = \\ = sigmoid(e^{k_0+k_1o_1+k_2o_2+k_3o_3}). \quad (3)$$

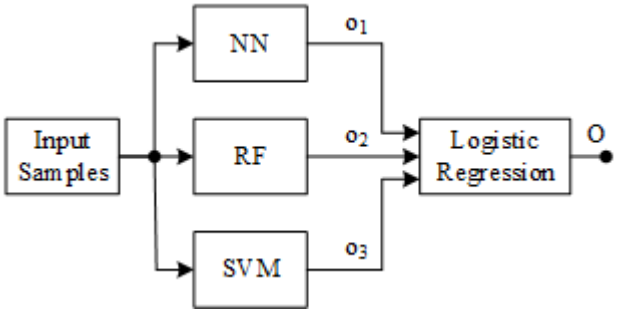


Figure 1. Structure of the proposed classifier

In general form, assuming a linear model, the following parametrized function could be defined:

$$g_{\vec{k}}(\vec{o}) = \frac{1}{1+e^{-\vec{k}^T\vec{o}}} = P(O = 1|\vec{o};\vec{k}), \quad (4)$$

where $\vec{k} = [k_0, k_1, k_2, k_3]$ and $\vec{o} = [1, o_1, o_2, o_3]$. Then, $P(O = 0|\vec{o};\vec{k}) = 1 - g_{\vec{k}}(\vec{o})$ and the probability for particular observation $o \in \{O\}$ is given by:

$$P(o|\vec{o};\vec{k}) = g_{\vec{k}}(\vec{o})^o\left(1 - g_{\vec{k}}(\vec{o})\right)^{(1-o)}. \quad (5)$$

The likelihood function then is expressed as:

$$\mathcal{L}(\vec{k}|o;\vec{o}) = P(O|\vec{o};\vec{k}) = \prod_i P(o_i|\vec{o}_i;\vec{k}) = \\ = \prod_i g_{\vec{k}}(\vec{o}_i)^{o_i}\left(1 - g_{\vec{k}}(\vec{o}_i)\right)^{(1-o_i)}. \quad (6)$$

The logarithm of L as likelihood function should be maximized using the gradient descent algorithm [18].

B. Combined multi-attack classifier

Classification of the input samples into 10 types of attacks and normal instances, that is o_1, o_2, o_3, $O \in \{0, 1, 2, \ldots, 10\}$ could be achieved by the single classifiers from Fig. 1 after combining them with the use of logistic regression of multinomial type.

Following the general approach from [17] and given an exemplary vector of outputs from the single classifiers $\vec{o}_i = [o_{1i}, o_{2i}, o_{3i}]$, obtained as a result from the *i*-th observation from the input, a prediction function of linear type could be defined, following:

$$S(\vec{o}_i, c) = \vec{w}_c \vec{o}_i, \tag{7}$$

where *c* is a current class, $c \in \{C\} \equiv \{0, 1, 2, \ldots, 10\}$ – one of the possible outcomes from classification, $\vec{w}_c$ – vector of weights, associated with the *c*-th class, and *S* is the score for the level of belonging of $\vec{o}_i$ to *c*.

The following expression holds true:

$$\begin{vmatrix} ln\frac{P(O_i=1)}{P(O_i=0)} = \vec{w}_1\vec{o}_i \\ ln\frac{P(O_i=2)}{P(O_i=0)} = \vec{w}_2\vec{o}_i \\ \vdots \\ ln\frac{P(O_i=10)}{P(O_i=0)} = \vec{w}_{10}\vec{o}_i \end{vmatrix}. \tag{8}$$

It is related to the fact, that the complete multinomial model with C possible output values could be built upon C-1 binary logistic regressions, which are independent among themselves. One of the resulting outputs is selected as a base and the rest C-1 are processed taking it as a reference. If the reference is c = 0, then we come to (8).

Putting the left and right side of (8) to exponent will eventually give:

$$\begin{vmatrix} P(O_i = 1) = P(O_i = 0)e^{\vec{w}_1\vec{o}_i} \\ P(O_i = 2) = P(O_i = 0)e^{\vec{w}_2\vec{o}_i} \\ \vdots \\ P(O_i = 10) = P(O_i = 0)e^{\vec{w}_{10}\vec{o}_i} \end{vmatrix} \tag{9}$$

and taking into account that all possible outputs form a full set, then:

$$P(O_i = 0) = 1 - \sum_{j=1}^{10} P(O_i = j) = \\ = 1 - \sum_{j=1}^{10} P(O_i = 0)e^{\vec{w}_j\vec{o}_i}. \tag{10}$$

From (10) it easy to find that:

$$P(O_i = 0) = \frac{1}{1+\sum_{j=1}^{10} P(O_i=0)e^{\vec{w}_j\vec{o}_i}}, \tag{11}$$

along with:

$$\begin{vmatrix} P(O_i = 1) = \frac{e^{\vec{w}_1\vec{o}_i}}{1+\sum_{j=1}^{10} P(O_i=0)e^{\vec{w}_j\vec{o}_i}} \\ P(O_i = 2) = \frac{e^{\vec{w}_2\vec{o}_i}}{1+\sum_{j=1}^{10} P(O_i=0)e^{\vec{w}_j\vec{o}_i}} \\ \vdots \\ P(O_i = 10) = \frac{e^{\vec{w}_{10}\vec{o}_i}}{1+\sum_{j=1}^{10} P(O_i=0)e^{\vec{w}_j\vec{o}_i}} \end{vmatrix}. \tag{12}$$

The components of the vectors $\vec{w}_j$, j = 1, 2, …, 10 could be found using he maximum a posteriori method [19]. Ranking the probabilities from (11)-(12) for every input $\vec{o}_i$ leads to the predicted type of attack or its absence.

The output of the NN is [13]:

$$o_{1i}^k = b_i^k + \sum_{j=1}^{N_k} v_{ji}^k o_j^{k-1} \tag{13}$$

where o_{1i}^k is the output of the *i*-th neuron from the output layer *k*; *k* = 3 for the detector and *k* = 9 for the classifier; b_i^k – the bias for the same neuron; o_j^{k-1} – the output for the *j*-th neuron from the previous layer (*k*-1); v_{ji}^k – the weight of neuron *i* from layer *k*, connecting it to neuron *j* from (*k*-1)-th layer; N_k – the number of neurons in layer *k*.

In the same time, the output from the RF could be found easily, taking into account the considerations of its generation, given in [20]:

$$o_2 = \sum_{i=1}^{n} \left(\frac{1}{m}\sum_{j=1}^{m} \omega_j(\vec{x}_i, \vec{x}')\right) y_i, \tag{14}$$

where $\vec{x}_i$ is the *i*-th vector from the training set for *i* = 1, 2, …, *n*; y_i – the associated label for $\vec{x}_i$; $\vec{x}'$ - the current input test sample to be classified; ω_j – weight function for the *j*-th decision tree from the forest after training the RF, *j* = 1, 2, …, *m*.

The last output o3, associated with the SVM, could be derived from the equations, describing its principle from [21]:

$$o_3 = \sum_{i=1}^{m} \alpha_i y^{(i)} K(\vec{x}^{(i)}, \vec{x}') + b, \tag{15}$$

where α_i is coefficient, associated with the *i*-th support vector $\vec{x}^{(i)}$, having a label $y^{(i)}$, *i* = 1, 2, …, *m*; *b* – a bias.

Substituting (13)-(15) in (5) for the combined detector and in (11)-(12) for the combined classifier leads to the decision function in its complete form for both of them.

III. Experimental results

A. Experimental setup

The test environment includes the following hardware components: 64-bit 4 core CPU Xeon E5-1620 with base frequency 3.5 GHz and 256 kB cache at level L1, 1 MB – at L2, and 10 MB – at L3, 64 GB of RAM and 7200 rpm 2 TB HDD. Implementation of all classifiers has been done within the Orange v. 3.28 machine learning application, running under the 64-bit MS Windows Pro 10 operating system.

The experimental dataset [16] includes 2934817 training samples with 370 of them gathered from normal traffic and 733705 test samples with 107 instances from normal traffic. The class (Cl.) for this type of samples is indexed as 0. The other attacks are: 1 – DoS TCP flood, 2 – DoS UDP flood, 3 – DoS HTTP flood, 4 – DDoS TCP flood, 5 – DDoS UDP, 6 – DDoS HTTP flood, 7 – Keylogging, 8 – Data Theft, 9 – OS Fingerprinting, 10 – Service Scanning. When making binary classification Cl. = 1 means an attack. The complete features contain 10 components: seq – number of the sequence for a record, stddev – standard deviation of records from particular type, N_IN_Conn_P_SrcIP – number of input connections for the attacking machine, , min – minimal duration time of a connection in the record, state_number – state of the record, mean – average time of connection for a record, N_IN_Conn_P_DstIP – number of input connections for the attacked machine, drate – packets per second from the attacked machine to the attacker, srate – packets per second from the attacking machine to the attacked one, max – maximal duration of connection for a particular record. As some of our previous research suggests [12, 13, 14, 15], this set of 10 features could be reduced to 8, omitting the seq and N_IN_Conn_P_SrcIP (being least informative), still preserving the classification accuracy of attacks in some cases. This is the motivation of trying the reduced set here as well in all the experiments, described below.

An optimal configuration of a feedforward neural network with back propagation, proposed in our previous study [12] and acting as a detector, consists of 60 neurons in a single hidden layer, when using 10 features and 80 neurons – for 8. The optimal configuration of the same type of network [13], operating as multi-attack classifier, consists of the following number of neurons in the hidden layers – 30-40-40-60-80-80-100 for both 8 and 10 features, employing the tangent hyperbolic activation function. All neural networks use the Adam training algorithm in no more than 1000 epochs, found to be sufficient to get the most accurate results, while the regularization parameter is $\alpha = 0.0001$.

From other earlier study of ours [14], an optimal configuration for the SVM is found as: Radial Basis Function (RBF) for the kernel, cost parameter C = 1, numerical tolerance *NT* = 10-3, valid for both the detector and classifier implementations. The iteration limit, tested to be enough for getting sufficient precision, is *IL* = 105 for the detector and 106 for the classifier.

Another preceding research, carried out over the same dataset by us [15] with a RF reveal that the optimal number of trees is 10 and the minimal number of subsets to split is 5.

All classifiers' configuration parameters from above are used within this study to evaluate the work of the combined classifiers (Fig. 1), denoted as NN+RF+SVM – operating once as detectors, discriminating malicious vs. normal traffic at 8 and 10 features, and then as classifiers of the 10 types of attacks along with the normal traffic samples, again at 8 and 10 features. Simplified configuration of the combined classifiers, excluding the SVM and retaining only the NN and RF, is also tested, given the same initialization conditions. These implementations are denoted as NN+RF.

The rate of correct classifications is being evaluated by the Area Under the Curve (AUC), Classification Accuracy (CA), F1-measure, Precision, Recall, Log-loss and Specificity parameters, which precise definitions could be found in [12].

B. Classification efficiency

Training, validation and testing times for all detectors are given in Table 1. Validation is made over the complete training set after the training process is over. This approach is also applied for the multi-attack classifiers.

Table 1. Detectors processing times

Detector	Features	Training Time, sec	Validation Time, sec	Test time, sec
NN+RF	8	1541.12	16.82	4.66
	10	1914.30	23.22	5.86
NN+RF+ SVM	8	412175.32	316.97	79.16
	10	197624.27	337.28	84.24

The training, validation and testing times for all investigated classifiers are presented in Table 2. Initial tests, when the iteration limit for the SVM is set to *IL* = 106, revealed extremely long training time at 10 features - 2169798.82 sec. Subsequent testing revealed that the overall classification accuracy is sustained at comparable levels when decreasing the *IL* to 1000 and all results from below are obtained with that tuning parameter for the SVM.

Table 2. Classifiers processing times

Detector	Features	Training Time, sec	Validation Time, sec	Test time, sec
NN+RF	8	40809.83	106.83	31.78
	10	80890.32	84.87	29.70
NN+RF+ SVM	8	123599.38	4935.23	1364.18
	10	195493.99	5020.27	1274.48

Detection accuracy for the 2-component detector at 8 features is given in Table 3 for both the validation over the training set and testing with unknown samples. The same procedure led to the results for the NN+RF detector, when employing 10 features, given in Table 4.

Table 3. NN+RF attack 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	0.9836	0.9917	0.9756	2.3153	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	2.3153	0.9756
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	2.3153	0.9756
Test	0	0.9999	0.9999	0.9509	1.0000	0.9065	5.7629	1.0000
	1	0.9999	0.9999	0.9999	0.9999	1.0000	5.7629	0.9065
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	5.7629	0.9065

Table 4. NN+RF attack detection efficiency using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, .10-5	Specifi-city
Train	0	0.9999	0.9999	0.9891	0.9945	0.9837	1.3824	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	1.3824	0.9837
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	1.3824	0.9837
Test	0	0.9999	0.9999	0.9463	0.9897	0.9065	8.7592	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	8.7592	0.9065
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	8.7592	0.9065

Adding the SVM to the combined detector change the accuracy, according to Table 5.

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, .10-5	Specifi-city
Train	0	0.9999	0.9999	0.9781	0.9889	0.9675	2.4250	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	2.4250	0.9675
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	2.4250	0.9675
Test	0	0.9999	0.9999	0.9463	0.9897	0.9065	8.0676	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	8.0676	0.9065
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	8.0676	0.9065

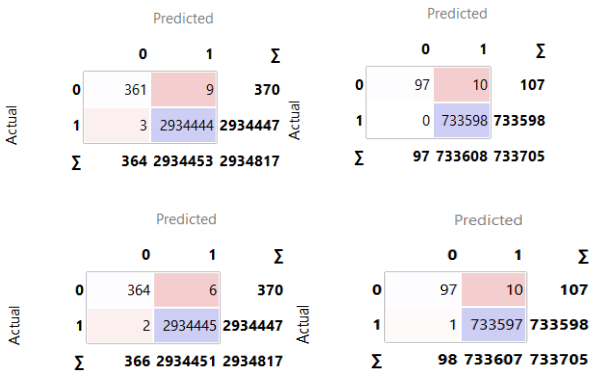
Table 5: NN+RF+SVM attack detection efficiency using 8 features

Increasing the number of components of the feature vector to 10 in the NN+RF+SVM detector result in discriminating rates as revealed by Table 6.

Table 6. NN+RF+SVM attack detection efficiency using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, .10-5	Specifi-city
Train	0	0.9999	0.9999	0.9932	0.9972	0.9891	1.2962	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	1.2962	0.9891
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	1.2962	0.9891
Test	0	0.9999	0.9999	0.9306	0.9894	0.8785	6.8347	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	6.8347	0.8785
	Av.	0.9999	0.9999	0.9999	0.9999	0.9999	6.8347	0.8785

Confusion matrices from validation over the training set and processing the test set for all detectors from the experimentation are given in Fig. 2.



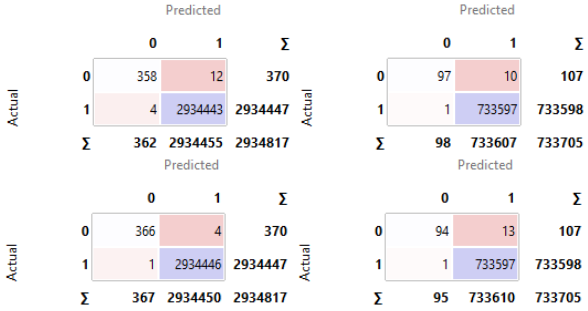


Figure 2.. Detectors confusion matrices: a – NN+RF train at 8 features, b – NN+RF test at 8 features, c – NN+RF train at 10 features, d – NN+RF test at 10 features, e - NN+RF+SVM train at 8 features, f - NN+RF+SVM test at 8 features, g – NN+RF+SVM test at 10 features, h – NN+RF+SVM test at 10 features

The accuracy rate from validation of the NN+RF implementation at 8 features is presented in Table 7.

Table 7. NN+RF train classification efficiency using 8 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, .10-5	Specifi-city
Training	0	0.9999	0.9999	0.9905	0.9892	0.9918	1.6283	0.9999
	1	0.9999	0.9992	0.9976	0.9971	0.9981	234.76	0.9994
	2	0.9999	0.9999	0.9999	0.9999	0.9999	1.5162	0.9999
	3	0.9999	0.9999	0.9970	0.9983	0.9957	1.4436	0.9999
	4	0.9999	0.9990	0.9981	0.9980	0.9982	375.05	0.9992
	5	0.9999	0.9999	0.9999	0.9999	0.9999	1.0906	0.9999
	6	0.9999	0.9999	0.9974	0.9961	0.9987	1.3483	0.9999
	7	0.9999	0.9999	0.8888	0.9795	0.8135	1.2793	0.9999
	8	0.9999	0.9999	N/A	N/A	N/A	1.0052	1.0000
	9	0.9968	0.9957	0.2908	0.8175	0.1768	888.81	0.9998
	10	0.9993	0.9958	0.9041	0.8348	0.9860	0.0086	0.9960
	Av.	0.9995	0.9990	0.9064	0.9610	0.8958	137.08	0.9994

Table 8 contains the accuracy related parameters for the same classifier, again at 8 features.

Table 8. NN+RF test classification efficiency using 8 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, .10-5	Specifi-city
Testing	0	0.9999	0.9999	0.9581	0.9537	0.9626	4.1419	0.9999
	1	0.9999	0.9988	0.9965	0.9962	0.9968	386.01	0.9992
	2	0.9999	0.9999	0.9999	0.9999	0.9999	6.2336	0.9999
	3	0.9999	0.9999	0.9866	0.9899	0.9833	4.4529	0.9999
	4	0.9999	0.9986	0.9974	0.9972	0.9976	517.63	0.9990
	5	0.9999	0.9999	0.9999	0.9999	0.9999	4.1026	0.9999
	6	0.9999	0.9999	0.9850	0.9949	0.9753	6.0413	0.9999
	7	0.9999	0.9999	0.6956	0.8888	0.5714	3.7190	0.9999
	8	1.0000	0.9999	N/A	N/A	N/A	0.7461	1.0000
	9	0.9960	0.9953	0.2319	0.6563	0.1408	990.47	0.9996
	10	0.9991	0.9954	0.8940	0.8239	0.9773	971.64	0.9957
	Av.	0.9994	0.9941	0.9928	0.9934	0.9941	1470.8	0.9995

Switching to 10 features as input to the NN+RF classifier change the validation results as shown in Table 9.

Table 9. NN+RF train classification efficiency using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, .10-5	Specifi-city
Training	0	0.9999	0.9999	0.9946	0.9919	0.9972	0.9392	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	1.4503	0.9999
	2	0.9999	0.9999	0.9999	1.0000	0.9999	1.0794	1.0000
	3	0.9999	0.9999	0.9995	1.0000	0.9991	0.8359	1.0000
	4	0.9999	0.9999	0.9999	0.9999	0.9999	1.5220	0.9999
	5	1.0000	1.0000	1.0000	1.0000	1.0000	0.6572	1.0000
	6	0.9999	0.9999	0.9993	0.9987	1.0000	0.6732	0.9999
	7	0.9999	0.9999	0.9473	0.9818	0.9152	1.0499	0.9999
	8	0.9999	0.9999	0.6666	1.0000	0.5000	0.4979	1.0000

	9	0.9999	0.9997	0.9780	0.9859	0.9701	73.688	0.9999
	10	0.9999	0.9997	0.9946	0.9927	0.9966	73.730	0.9998
	Av.	0.9999	0.9998	0.9617	0.9955	0.9434	17.016	0.9999

The test set yields classification rate in the same configuration as the above, presented in Table 10.

Table 10. NN+RF test classification efficiency using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Testing	0	0.9999	0.9999	0.9619	0.9805	0.9439	3.1206	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	8.7560	0.9999
	2	0.9999	0.9999	0.9999	0.9999	0.9999	4.8040	0.9999
	3	0.9999	0.9999	0.9900	0.9900	0.9900	2.5594	0.9999
	4	0.9999	0.9999	0.9999	0.9999	0.9999	8.9424	0.9999
	5	0.9999	0.9999	0.9999	0.9999	0.9999	2.0062	0.9999
	6	0.9999	0.9999	0.9877	0.9852	0.9901	2.1907	0.9999
	7	0.9999	0.9999	0.9230	1.0000	0.8571	1.2890	1.0000
	8	0.9999	1.0000	1.0000	1.0000	1.0000	0.3904	1.0000
	9	0.9998	0.9992	0.9212	0.9456	0.8980	207.13	0.9997
	10	0.9999	0.9992	0.9806	0.9745	0.9869	209.16	0.9994
	Av.	0.9998	0.9991	0.9991	0.9991	0.9991	227.82	0.9999

The 3-component NN+RF+SVM classifier, using 8 features, produce validation results, observable in Table 11.

Table 11. NN+RF+SVM train classification efficiency using 8 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Training	0	0.9999	0.9999	0.9811	0.9784	0.9837	1.9083	0.9999
	1	0.9999	0.9992	0.9976	0.9972	0.9980	233.38	0.9994
	2	0.9999	0.9999	0.9999	0.9999	0.9999	1.5891	0.9999
	3	0.9999	0.9999	0.9966	0.9983	0.9949	1.5730	0.9999
	4	0.9999	0.9990	0.9981	0.9980	0.9982	368.36	0.9992
	5	0.9999	0.9999	0.9999	0.9999	0.9999	1.3275	0.9999
	6	0.9999	0.9999	0.9949	0.9949	0.9949	2.1150	0.9999
	7	0.9999	0.9999	0.9491	0.9491	0.9491	0.9478	0.9999
	8	0.9999	0.9999	N/A	N/A	N/A	0.7558	1.0000
	9	0.9969	0.9957	0.2890	0.8129	0.1757	882.05	0.9998
	10	0.9993	0.9958	0.9038	0.8346	0.9856	862.98	0.9960
	Av.	0.9975	0.9948	0.9937	0.9947	0.9948	1192.9	0.9996

In the same time, testing with unknown samples changes these values as noted in Table 12.

Table 12. NN+RF+SVM test classification efficiency using 8 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Testing	0	0.9999	0.9999	0.9668	0.9807	0.9532	5.1888	0.9999
	1	0.9999	0.9988	0.9967	0.9961	0.9973	370.95	0.9992
	2	0.9999	0.9999	0.9999	0.9999	0.9999	5.8489	0.9999
	3	0.9999	0.9999	0.9867	0.9867	0.9867	4.5821	0.9999
	4	0.9999	0.9986	0.9975	0.9975	0.9975	499.21	0.9991
	5	0.9999	0.9999	0.9999	0.9999	0.9999	3.5179	0.9999
	6	0.9998	0.9999	0.9775	0.9898	0.9655	7.6863	0.9999
	7	0.9999	0.9999	0.7500	0.9000	0.6428	4.2446	0.9999
	8	1.0000	0.9999	N/A	N/A	N/A	0.5183	1.0000
	9	0.9960	0.9953	0.2219	0.6607	0.1333	0.0096	0.9996
	10	0.9991	0.9954	0.8943	0.8232	0.9789	947.85	0.9957
	Av.	0.9994	0.9941	0.9928	0.9934	0.9941	1431.2	0.9995

The final, most comprehensive configuration - NN+RF+SVM classifier with 10 features has validation accuracy as shown in Table 13.

Table 13. NN+RF+SVM train classification efficiency using 10 features

Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Trainin g	0	0.9999	0.9999	0.9866	0.9736	1.0000	1.0956	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	1.6971	0.9999
	2	0.9999	0.9999	0.9999	0.9999	0.9999	1.1746	0.9999
	3	1.0000	1.0000	1.0000	1.0000	1.0000	0.7480	1.0000
	4	0.9999	0.9999	0.9999	0.9999	0.9999	1.5309	0.9999
	5	1.0000	0.9999	0.9999	1.0000	0.9999	0.6542	1.0000
	6	1.0000	0.9999	0.9987	1.0000	0.9974	0.6026	1.0000
	7	0.9999	0.9999	0.9464	1.0000	0.8983	0.8923	1.0000
	8	0.9999	0.9999	0.7272	0.8000	0.6666	0.3817	0.9999
	9	0.9999	0.9997	0.9777	0.9866	0.9690	74.302	0.9999
	10	0.9999	0.9997	0.9946	0.9924	0.9967	74.371	0.9998
	Av.	0.9999	0.9997	0.9997	0.9997	0.9997	79.146	0.9999

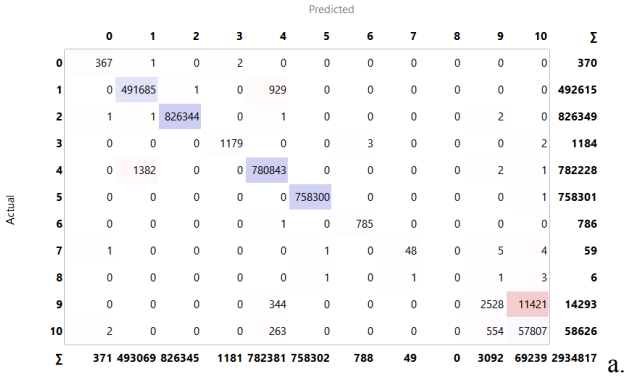
Its accuracy, being tested with unknown samples, could be observed in Table 14.

Table 14. NN+RF+SVM test classification efficiency using 10 features

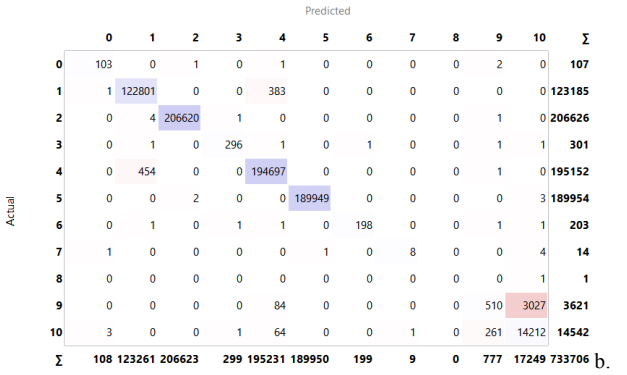
Set	Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss, $.10^{-5}$	Specifi-city
Testing	0	0.9999	0.9999	0.9532	0.9532	0.9532	4.3080	0.9999
	1	0.9999	0.9999	0.9999	0.9999	0.9999	9.7101	0.9999
	2	0.9999	0.9999	0.9999	0.9999	0.9999	5.8957	0.9999
	3	0.9999	0.9999	0.9933	0.9966	0.9900	2.0255	0.9999
	4	0.9999	0.9999	0.9999	0.9999	0.9999	8.9609	0.9999
	5	0.9999	0.9999	0.9999	0.9999	0.9999	2.5380	0.9999
	6	0.9999	0.9999	0.9950	0.9950	0.9950	1.9193	0.9999
	7	1.0000	0.9999	0.9230	1.0000	0.8571	1.0250	1.0000
	8	1.0000	1.0000	1.0000	1.0000	1.0000	0.2286	1.0000
	9	0.9998	0.9991	0.9153	0.9390	0.8928	210.45	0.9997
	10	0.9999	0.9991	0.9794	0.9734	0.9855	211.18	0.9994
	Av.	0.9999	0.9991	0.9991	0.9991	0.9991	232.02	0.9999

In Tables 7-14, whenever N/A appears as a resulting value for some of the parameters, it is related to inability to calculate this value with enough precision.

Confusion matrices from classification with NN+RF are shown in Fig. 3.



a.



b.

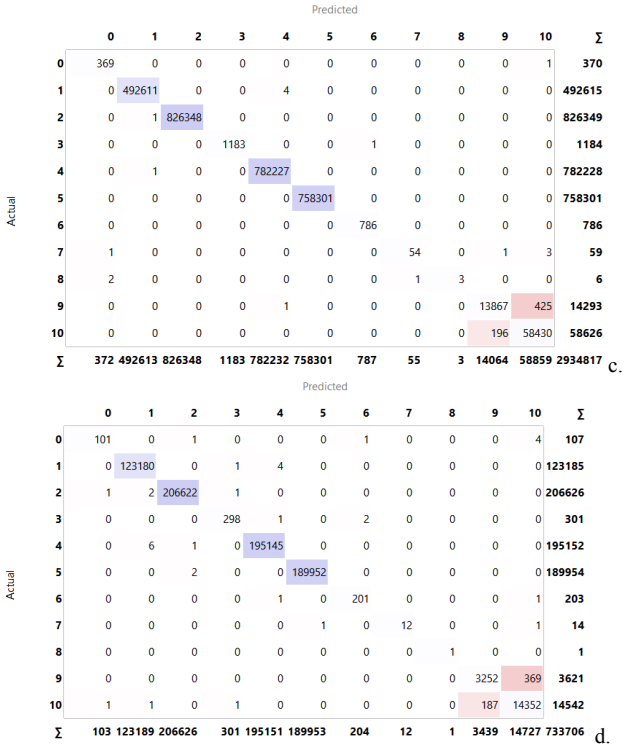
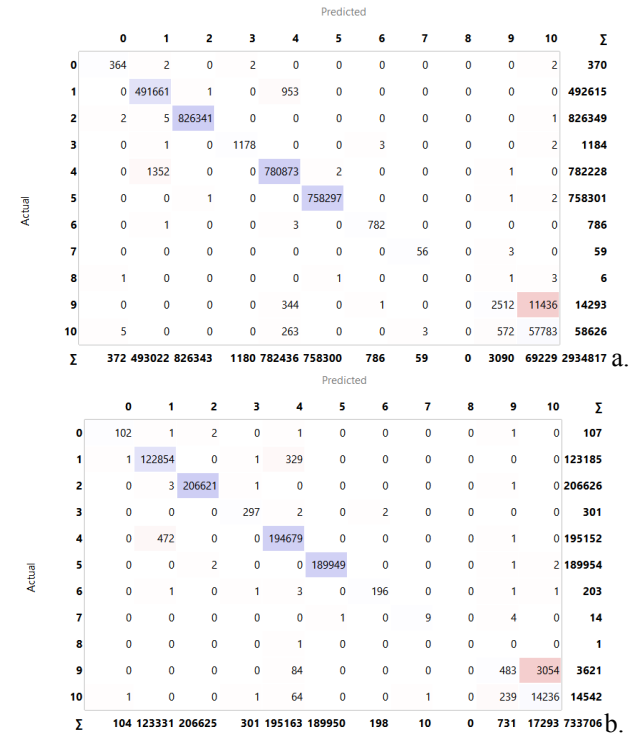


Figure 3. NN+RF classifiers confusion matrices from: a – train set at 8 features, b – test set at 8 features, c – train set at 10 features, d – test set at 10 features

Fig. 4 depicts the confusion matrices from classification with NN+RF+SVM. The experiment here is also implemented in two ways – first, using only the 8-component feature vectors for both the validation and testing, and then repeating this procedures with the 10-component vectors.



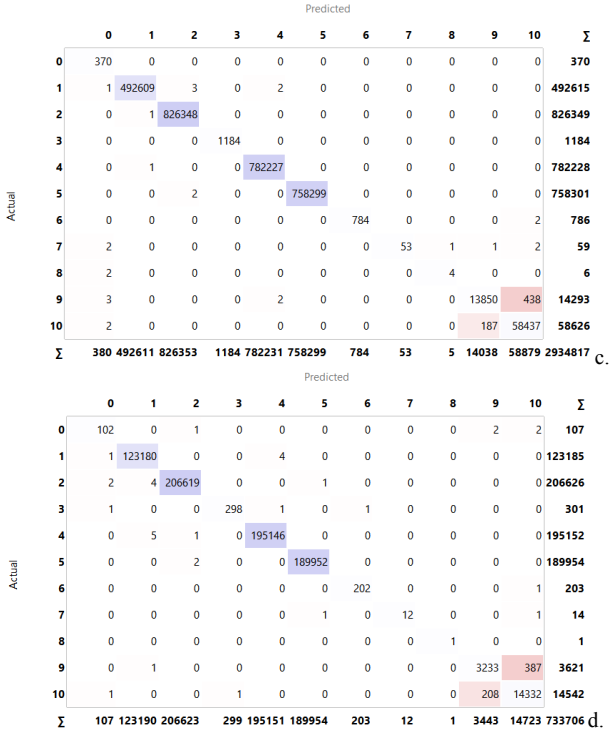


Figure 4. NN+RF+SVM classifiers confusion matrices from: a – train set at 8 features, b – test set at 8 features, c – train set at 10 features, d – test set at 10 features

IV. Discussion

The most accurate of all combined detectors turns to be the NN+RF, using 8 features with only 10 wrongly classified non-attack samples as attacks (Fig.2.b). Testing time takes 4.66 sec (Table 1). All the other 3 configurations seem to be very close by classification accuracy – 13 erroneously classified non-attacks and 1 attack being missed by the NN+RF+SVM, using 10 features and being the most inaccurate detector (Fig. 2.h). Training, validation and testing times for the NN+RF rise 1.24, 1.38 and 1.26 times, respectively at 10 features, compared to 8. There is a drop in training time by 2.09 times at 10 features, related to 8, for the NN+RF+SVM, while the validation and testing times increase by 1.06 times. This result could be explained by the longer process it takes for the SVM to reach targeted accuracy when being fed with the less informative 8-feature set. The optimal configuration of the feedforward neural network (NN) alone from our previous study [12] yields errors for 9 non-attack and 27 attack samples for 2.45 sec. Optimal single RF [14] classify incorrectly 4 attack and 2 attack samples in 6.21 sec. If a particular DDoS prevention, already using a NN of the kind presented here needs to be improved, then it could take as a second classifier the RF. If such a system should be design from scratch it is more appropriate to use a single RF detector. All these results are additionally supported by the accuracy parameters, given in Tables 3-6.

Classification over the 10 types of attacks, using the NN+RF classifier, for both 8- and 10-feature implementations takes almost the same time of around 100 sec for the validation

process and around 30 sec for processing the test set (Table 2). Similar is the case for the NN+RF+SVM in its two variants with around 5000 sec for validation and around 1300 sec for testing, visible from the same table. Training time rises for both NN+RF and NN+RF+SVM, when switching from 8 to 10 features. The increase is 1.58 and 1.98 times, respectively (Table 2). Discrimination in more than 2 classes does not imply longer training for the reduced set of features, when the SVM is part of the combined classifier, as it is the case with the detector. The most accurate on average from all handled attacks and normal traffic appears the NN+RF+SVM classifier, using 10 features (Fig. 4.d) with 97.03% correctly classified samples. It is very tightly followed by the NN+RF with 10 features, achieving 96.96% (Fig. 3.d). For most of the attacks both classifiers are very close in accuracy, diverging one from the other by a few samples (from 1 to 3) and only for the OS Fingerprinting (9) and Service Scanning (10) the difference is 19 and 20 samples in flavor for the NN+RF classifier. Given the smaller training time with a factor of 2.42 and the more significant difference of 42.91 times less for the test time of NN+RF, compared to NN+RF+SVM (Table 2), it could be recommended the use of NN+RF at 10 features in the general case for discovering the types of attacks under consideration in this study. Deeper analysis for this classifier (Fig. 3.d) reveals that mostly (at least 0.1% difference) mismatching of non-attack samples happens as samples of DoS UDP flood (0.9%), DDoS HTTP flood (0.9%), and Service Scanning (3.7%); for DoS HTTP flood – with samples as DDoS TCP flood (0.3%) and DDoS HTTP flood (0.7%); for DDoS HTTP flood – with samples as DDoS TCP flood (0.5%) and Service Scanning (0.5%); for Keylogging – as DDoS UDP (7.1%) and Service Scanning (7.1%); for OS Fingerprinting – as Service Scanning (10.2%); and for Service Scanning – as OS Fingerprinting (1.3%). Further development of the combined classifier, possibly by extending it with another type of a single classifier, could further reduce the mismatching rate among listed attacks. All these observations are additionally supported by the values of accuracy parameters from Tables 7-14. Comparison by the relative number of correctly detected samples from the test set with our previous study on the RF single classifier [15] over the same dataset, proven to be more accurate than the NN [13] and the SVM [14], which we also tested, is given in Table 15.

Table 15. Accuracy comparison, in %, between the optimal NN+RF, proposed here, and the RF, from [15], classifiers

Attack	0	1	2	3	4	5
NN+RF	94.4	100.0	100.0	99.0	100.0	100.0
RF, [15]	91.6	100.0	100.0	99.7	100.0	100.0
Attack	6	7	8	9	10	-
NN+RF	99.0	85.7	100.0	89.8	98.7	-
RF, [15]	99.0	92.9	0.00	91.2	98.4	-

One of the major advantages of the combined classifier, proposed here is the higher rate of detected non-attack samples (Table 15). The RF [15] process the same amount of test samples for 12.69 sec, while for the NN+RF it takes 29.70 sec. If the false alarm rate is not crucial for the particular application, the RF alone could also be employed.

Another comparison, related to the accuracy of the proposed here optimal detector NN+RF at 8 features with the accuracy of 3 other types of detectors, proposed by other authors and tested with the same dataset [16], is given in Table 16

Table 16. Accuracy comparison of attack detectors

Detector	SVM, [16], 10 features	RNN, [16], 10 features	LSTM, [16] 10 features	Proposed NN+RF, 8 features
CA	0.8837	0.9974	0.9974	0.9999
Precision	1.0000	0.9999	0.9999	0.9999
Recall	0.8837	0.9975	0.9975	0.9999

The classification efficiency of the NN+RF implementation from this study, working with 10 features, is being compared with an RNN classifier [16], tested over the same dataset. The results are given in Table 17. The Classification Accuracy (*CA*) parameter is being equal for the DDoS TCP, DDoS UDP, DoS HTTP, DoS TCP, and DoS UDP attacks for both classifiers. NN+RF, using 10 features, has higher *CA* for the DDoS HTTP, DoS HTTP, OS Fingerprinting, Service Scan, Data Exfiltration, and Keylogging by 0.67%, 1.93%, 0.75%, 0.35%, 1.24%, and 1.26%, respectively. Nevertheless, some of the values for *Precision* and *Recall* are smaller for the proposed here classifier, such as those for DDoS HTTP, OS Fingerprinting, Service Scan, and Keylogging (just for the *Recall*) attacks. Taking into account the confusion matrix from Fig. 3.d, it could be investigated further the distribution of False Negatives along the rows of the respective attacks. Then, observing in detail the components of the feature vectors for these samples as mutual distributions with the components of features from other attacks, it could bring as a hint possible extension of the employed features for further improvement of those 2 parameters – *Precision* and *Recall*, for this particular classes. The current overall performance of the proposed in this paper classifier is considered high enough, both from computational standpoint and as an accurate mean to discriminate some of the most typical IoT-based network attacks.

Table 17. Accuracy comparison of attack classifiers

Attack	DDoS HTTP		DDoS TCP	
	RNN, [16]	Proposed	RNN, [16]	Proposed
CA	0.9932	0.9999	0.9999	0.9999
Precision	0.9930	0.9852	0.9999	0.9999
Recall	0.9970	0.9901	0.9999	0.9999
Attack	DDoS UDP		DoS HTTP	
	RNN, [16]	Proposed	RNN, [16]	Proposed
CA	0.9999	0.9999	0.9806	0.9999
Precision	0.9999	0.9999	0.9898	0.9900
Recall	0.9999	0.9999	0.9845	0.9900
Attack	DoS TCP		DoS UDP	
	RNN, [16]	Proposed	RNN, [16]	Proposed
CA	0.9999	0.9999	0.9999	0.9999
Precision	0.9999	0.9999	0.9999	0.9999
Recall	0.9999	0.9999	0.9999	0.9999

Attack	OS Fingerprinting		Service Scan	
	RNN, [16]	Proposed	RNN, [16]	Proposed
CA	0.9917	0.9992	0.9957	0.9992
Precision	0.9984	0.9456	0.9986	0.9745
Recall	0.9931	0.8980	0.9971	0.9869
Attack	Data exfiltration		Keylogging	
	RNN, [16]	Proposed	RNN, [16]	Proposed
CA	0.9876	1.0000	0.9873	0.9999
Precision	0.0000	1.0000	0.9853	1.0000
Recall	0.0000	1.0000	0.9178	0.8571

V. Conclusion

In this paper 4 new models of binary classifiers of IoT-based network attacks are proposed. The first is a 2-link detector, discriminating normal traffic samples from 10 types of attacks samples (DoS and DDoS kinds of TCP, UDP and HTTP floods, Keylogging, Data Theft, OS Fingerprinting, and Service Scanning). It is comprised of feedforward neural network with 1 hidden layer and a random forest, connected through a logistic regression. The second is 3-link detector – with an additional support vector machine. One pair of detectors operate over 8 feature vectors, and the other pair – over 10 feature vectors - with 2 new different components from the first. The error rate of the 2-link detector at 8 features is $1.36.10^{-3}$%, and $1.50.10^{-3}$% at 10 features. The 3-link detector has $1.50.10^{-3}$% error rate at 8 features and $1.91.10^{-3}$% - at 10 features. The errors are mostly concentrated in the non-attack samples. The neural network alone, as our previous study suggests, has an error rate of $4.91.10^{-3}$% in its optimal configuration as a detector, working with 10 features. From another 2 preceding studies of ours, it is estimated that the support vector machine introduces $9.81.10^{-3}$% errors, while the random forest has $0.82.10^{-3}$% errors, working with 8 features as a detector. The errors, induced by the single detectors, are more evenly scattered between attack and non-attack samples. Training time of the 3-link detector, compared to the optimal 2-link, when using 8 features takes 267.5 times longer. Testing time ratio between the same 2 configurations is almost 17 times. The 2-link detector at 8 features prove to be the optimal detector among all tested combined binary classifiers in this study.

The same combination of single classifiers through linear regression as 2- and 3-links classifiers, working on 8- and 10-feature sets are tested. They give as an output 1 of 11 possible values – 0 for predicted non-attack samples and 1-10 – for 1 of 10 possible attacks. The 2-link classifier at 8 and 10 features yields 78.23% and 96.96%, respectively, correctly classified samples of the total number of test samples. The 3-link classifier at 8 and 10 features has 78.69% and 97.04%, respectively, correct predictions over the test set. In the same time, as our previous research reveals, the optimized support vector machine (10 features) alone achieves 62.62% correctly classified samples, and an optimized neural network at 10 features – 76.35%. The random forest correct classifications are 97.28%. The more accurate 3-link classifier is 2.42 times slower than the very close as accuracy 2-link implementation.

Both optimal configurations of the combined detector and classifier could be useful in real practice.

References

[1] Hamid, H., Noor, R. M., Omar, S. N., Ahmedy, I., Anjum, S. S., Shah, S. A. A., Kaur, S., Othman, F., Tamil, E. M., IoT-based Botnet Attacks Systematic Mapping Study of Literature. *Scientometrics*, Vol. 126, No. 4, 2021, pp. 2759-2800.

[2] Koay, A., Chen, A., Welch, I., Seah, W. K., A New Multi Classifier System using Entropy-based Features in DDoS Attack Detection. *In 2018 International Conference on Information Networking (ICOIN)*, IEEE, January 2018, pp. 162-167.

[3] Das, S., Mahfouz, A. M., Venugopal, D., Shiva, S., DDoS Intrusion Detection through Machine Learning Ensemble. *In 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, IEEE, July 2019, pp. 471-477.

[4] Musumeci, F., Ionata, V., Paolucci, F., Cugini, F., Tornatore, M. Machine-learning-assisted DDoS Attack Detection with P4 Language. *In ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, June 2020, pp. 1-6.

[5] Mahfouz, A., Abuhussein, A., Venugopal, D., Shiva, S., Ensemble Classifiers for Network Intrusion Detection Using a Novel Network Attack Dataset. *Future Internet*, Vol. 12, No. 11, 2020, Art. No. 180.

[6] Algelal, Z. M., Aldhaher, E. A. G., Abdul-Wadood, D. N., Al-Sagheer, R. H. A., Botnet Detection using Ensemble Classifiers of Network Flow. *International Journal of Electrical and Computer Engineering*, Vol. 10, No. 3, 2020, pp. 2543-2550.

[7] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., Alazab, A., A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks, *Electronics*, Vol. 8, No. 11, 2019, Art. No. 1210.

[8] Rajagopal, S., Kundapur, P. P., Hareesha, K. S., A Stacking Ensemble for Network Intrusion Detection using Heterogeneous Datasets. *Security and Communication Networks*, Vol. 2020, Art. No. 4586875, 2020.

[9] Iwendi, C., Khan, S., Anajemba, J. H., Mittal, M., Alenezi, M., Alazab, M., The Use of Ensemble Models for Multiple Class and Binary Class Classification for Improving Intrusion Detection Systems, *Sensors*, Vol. 20, No. 9, 2020, Art. No. 2559.

[10] Jain, A. K., Dhawan, H., Sowmiya, B., DDoS Detection Using Machine Learning Ensemble. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, Vol. 12, No. 12, 2021, pp. 1647-1655.

[11] Zhou, Y., Cheng, G., Jiang, S., Dai, M., Building an Efficient Intrusion Detection System based on Feature Selection and Ensemble Classifier. *Computer Networks*, Vol. 174, 2020, Art. No. 107247.

[12] Ivanova, V., T. Tashev, I. Draganov, Detection of IoT based DDoS Attacks by Network Traffic Analysis using

Feedforward Neural Networks, WSEAS, 2021 (under review).
[13] Ivanova, V., Multiple IoT based Network Attacks Discrimination by Multilayer Feedforward Neural Networks, WSEAS, 2021 (under review).
[14] Ivanova, V., T. Tashev, I. Draganov, DDoS Attacks Classification using SVM, WSEAS, 2021 (under review).
[15] Ivanova, V., T. Tashev, I. Draganov, Random Forest Detector and Classifier of Multiple IoT-based DDoS Attacks, WSEAS, 2021 (under review).
[16] 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.
[17] Hosmer Jr, D. W., Lemeshow, S., Sturdivant, R. X., *Applied Logistic Regression*, 3 ed., John Wiley & Sons, 2013.
[18] Kidambi, R., *Stochastic Gradient Descent for Modern Machine Learning: Theory, Algorithms and Applications*, PhD Thesis, University of Washington, 2019.
[19] Campisi, P., Egiazarian, K., *Blind Image Deconvolution: Theory and Applications*, CRC Press, 2007.
[20] Pavlov, Y. L., *Random Forests,* De Gruyter, 2019.
[21] Steinwart, I., Christmann, A., *Support Vector Machines*, Springer, 2008.
[22] (Journal Online Sources style) K. Author. (year, month). Title. Journal [Type of medium]. Volume(issue), paging if given. Available: http://www.(URL)
[23] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. IEEE Trans. Plasma Sci. [Online]. 21(3). pp. 876—880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar