DDoS Attacks Classification using SVM

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

Abstract: - In this paper two types of classifiers of Distributed Denial of Service (DDoS) attacks, based on Support Vector Machines, are presented – a binary and a multiclass one. They use numerical samples, aggregated from packet switched network connections records, captured between attacking machines, most typically IoT bots and a victim machine. Ten of the most popular DDoS attacks are studied and represented as either 10- or 8-feature vectors. Detection rate and classification accuracy is being measured in both cases, along with lots of other parameters, such as Precision, Recall, F1-measure, training and testing time, and others. Variations with Linear, Polynomial, RBF and Sigmoid kernels are being tried with the SVM. The most accurate turns out to be the RBF SVM, both as detector and multiclass classifier, achieving classification accuracy as high as 0.9999 for some of the attacks. Testing times reveal the practical fitness of the implemented classifiers for real-world application.

Key-Words: - distributed denial of service, network attack, Internet of Things, Support Vector Machine, kernel function, optimized classifier

Received: March 2, 2021. Revised: December 3, 2021. Accepted: January 15, 2022. Published: February 9, 2022.

1 Introduction

DDoS attacks induce large financial losses [1] by interrupting mass-type services, causing data loss and sometimes ease the process of compromising various Internet based machines, leading to data theft and other malicious activities. One of the more recent challenges, related to the prevention of DDoS attacks, is connected to the Software Defined Networks (SDN) and the ways of efficiently limiting their scale extension, analyzing the influence over the SDN controller [2]. Neural networks are not considered practical solution for this particular instance as Ye et al. [2] show, while employing 6 component features from the flow of the switch. They use them as training samples for Support Vector Machines (SVM) and get 95.24% accuracy of detecting DDoS attacks.

Detection and classification of DDoS attacks, along with their further prediction, has been proposed by Yusof et al. [3], using K-Nearest Neighbor as a combination with SVM (KNN-SVM). Good partitioning of the pre-attack data flow and the actual influence over the network with a peak traffic and other effects have been achieved. The proposed approach is considered useful for intrusion detection as well. The classification rate is in the order of 96.4% for the SVM and 96.6% for the KNN when using the KDD99 dataset. In [4] Daneshgadeh et al. combine Kernel Online Anomaly Detection, SVM and principles from the Information Theory with the aim of differentiating DDoS and Flash Events, the latter being completely legitimate activities. The authors also find their scheme useful for normal traffic discrimination. Another study, performed by Khuphiran et al. [5] gives a comparison between Deep Feed Forward (DFF) and SVM models, which are able of detecting DDoS attacks. Testing over DARPA 2009 dataset with these models, produce an accuracy of 99.63% for the DFF and 93.01% - for the SVM. DFF appears to be around 1.28 times faster than the SVM as per the training phase. Despite this relation, the authors report that SVM is faster during the testing phase and should be preferred if the accuracy is not the primary issue.

More general approach is undertaken in [6], where Ali et al. propose a framework, based on machine learning, directed towards the prevention of the DDoS attacks in SDN and in the same time it is capable of reducing the dimensionality of processed data, transferred in huge amounts. Thus, it becomes possible to reduce the risk of launching a spoof controller and changing the routing tables. Principal Component Analysis (PCA) is also implemented in the framework, along with SVM,

which goes as successful solution towards smaller false positive rate, and the overall accuracy increased. Another more complex approach, relying on a hybrid algorithm for spotting DDoS attacks is developed by Adhikary et al. [7]. It is specifically oriented towards Vehicular Adhoc Networks (VANETs). Kernel methods, based on AnovaDot and RBFDot, lie at the base of a SVM for solving that task. With regard to the normal traffic lots of real-world acting factors, e.g. packets loss, jitter and collisions, are being introduced to make the testbed as close to real networks as possible when the time comes to discriminate a DDoS attack from that normal traffic. AnovaDot and RBFDot realizations seem to be more efficient when combined than when applied independently.

During the recent years, it was also demonstrated that simpler realizations of SVM classifiers, such as the linear l_1 type, developed by Nazih et al. [8], could be efficiently exploited for discovering of attacks in Session Initiation Protocol (SIP) Voice over Internet Protocol (VoIP) networks. Denial of Service (DoS) and Spam over Internet Telephony (SPIT) attacks have been successfully discovered due to the introduction of n-gram string features, projected in a space with high dimensionality. Detection speed is higher for the l_1-SVM when discovering SPIT attacks, compared to the combined classifier, comprised of Markov Chain and SVM in the order of 20 times and in the same time the F1 measure for the proposed classifier is close to 100%, while the accuracy rate for the combined model is 96.3%. The l_1-SVM has almost the same accuracy into discovering DoS attacks with comparison to the Dual SVM and also considering the Malformed Msgs when compared to the SDP Parser.

Malware and spoofing discovery, along with DDoS attacks, is an object of the study, presented in [9] by Kajal and Nandal. Feature selection is done with the use of Genetic Algorithm as a first step, refining them later by Artificial Bee Colony and Discrete Wavelet Transform algorithms. Combining in a hybrid approach an Artificial Neural Network (ANN) and SVM allows the more accurate detection of malicious behaviors of separate nodes in the communication network. The increase in precision and recall while detecting DDoS attacks, compared to earlier strategy of query expansion with convolution kernels and dependency parses, is 0.112 and 0.049, respectively. The applicability of SVM into discovering DDoS network attacks, consisting primarily from HTTP (Hypertext Transfer Protocol) flood and DDoS using SQL (Structured Query Language) injection (SIDDoS) has been investigated in [10]. Multiple classifiers for the same task have been compared, such as Naïve Bayes, Decision Trees and Multilayer Perceptron (MPL). It appears that an Enhanced Multi Class SVM (EMCSVM) could detect accurately enough DDoS related events, while maintaining low false alarm rate, when taking as an input 14 components in a feature vector and trying to discriminate 10 kinds of attacks.

Based on all recent developments, as described above, in the field of detection and classification of DDoS attacks with the independent or combined use of SVM, we seek to find within this study the most optimal configuration of a single SVM, capable of detecting the presence of at least 1 type of such attack, discriminating it from normal traffic. Then, in a second, extended configuration, an SVM that could classify 1 of 10 kinds of attacks precisely enough, given the contemporary efficiency of comparable classifiers. Consideration has been made on the kernel function and configuration parameters while taking as an input 10- and 8-elements (reduced set) features from a recent and popular UNSW (University of New South Wales, Canberra, Australia) IoT-based (Internet of Things) DDoS attacks dataset [11].

In the Second part of the paper a brief description has been given of the distribution of features, depending on the present attack in the test dataset, as well as the general mathematical description of an SVM, using various kernel functions and their related parameters, and general optimization procedure to get the most efficient configuration of a classifier of this type. In Section 3, experimental results are given from testing SVMs as detectors and classifiers of DDoS attacks, revealing the most optimal configuration, followed by a discussion in Section 4. Finally, a conclusion is presented in part 5 of the paper.

2 Classifier Structure

2.1 Dataset and Feature Distribution

The dataset [11], used in this study, which is freely available and used by other authors in their research, contains 2934817 training and 733705 test samples. Each sample has 10 components, that are being aggregated from IP network flows – packets rate between source and destination machine (*srate*), the rate in opposite direction (*drate*), the estimated variance by its square root of the number of recorded connections (*stddev*), the consecutive number of a captured sequence (*seq*), the minimal, average and maximal period of exchange (*min*,

mean, *max*), identifier of a state for a feature (*state_number*), and the number of connections that get into the destination and the source machine (*N_IN_Conn_P_DstIP*, *N_IN_Conn_P_SrcIP*). All the feature components are represented as numerical values and the target variables are 2 kinds – a binary one with value 0 (non-attack) and 1 (attack) for testing the binary SVM classifier (attack detector); and numerical one with values 0 (non-attack), 1 (DoS TCP attack), 2 (DoS UDP attack), 3 (DoS HTTP attack), 4 (DDoS TCP attack), 5 (DDoS UDP attack), 6 (DDoS HTTP attack), 7 (Keylogging), 8 (Data Exfiltration), 9 (OS Fingerprint), 10 (Service Scan) for testing the SVM with multiclass outputs. Within the training set the number of non-attack records is 370, and in the test set – 107.

All the records are gathered from internal network setup with 4 simulated IoT devices, acting as bots and generating malicious traffic, corresponding to the attacks, described above. For this purpose the Kali Linux is used on conventional machines, a workstation with Windows 7, Mobile station and a Server under the control of Ubuntu operating systems play the role of attacked machines. The information of established connections is recorded by a separate monitoring station, connected to the switch, through which all other machines are communicating.

Ranking of the features based on their informative significance, related to the various classes separability, is being performed with the use of the χ^2 parameter, according to [14]:

$$\chi^2_{D_f} = \sum_{i=1}^{N} \frac{(O_i - E_i)^2}{E_i}, \quad (1)$$

where D_f is the number of the degrees of freedom, N – the sample size, O_i – observed values, E_i – expected values for $i = 1, 2, \ldots, N$. The number of degrees of freedom is associated with the number of independent values by any logical connection, that is they very independently one from another. Most often, it is true that [14]:

$$D_f = N - 1, \quad (2)$$

where 1 represents the number of constraints, being introduced independently when gathering all the samples during testing. In that case, the random variable χ will correspond to χ^2-distribution and it will be true that it is equal to the superposition of a number of variables, e.g. M, following a normal distribution [14]:

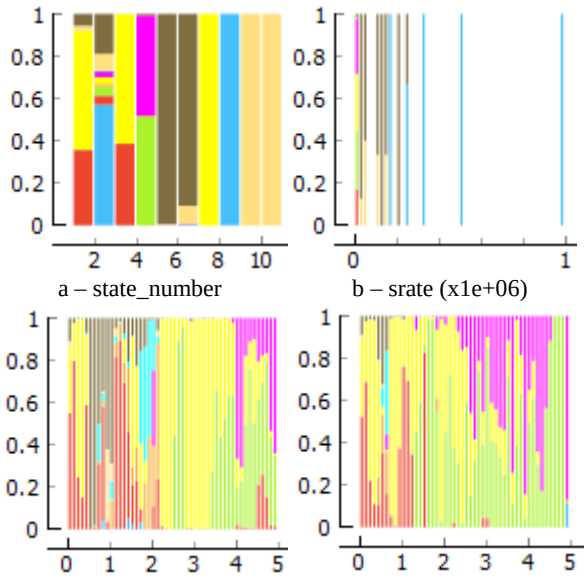
$$\chi^2 = \sum_{i=1}^{M} x_i^2. \quad (3)$$

The results are given in Table 1.

Table 1. χ^2 values for all features

mean	1357708.10	drate	621785.11
srate	1324153.22	min	515194.56
max	1168444.70	N_IN_Conn_P_DstIP	192709.72
state_number	1121064.98	seq	121856.09
stddev	666877.36	N_IN_Conn_P_DstIP	23049.61

The probability distribution each feature to be connected to a particular attack, being present with a given value, is shown in Fig. 1. The first 6 cases (Fig. 1 a-f) clearly indicate relatively good separability among the attacks over the range of these features – *state_number*, *srate*, *max*, *mean*, *stddev* and *min*. For the *N_IN_Conn_P_DstIP* the 1st, 2nd and 10th attack are covered altogether with their distributions almost over the entire range of this feature, and also significant portions of that range is covered of the distributions for other attacks. For *drate*, the 9th and 10th attack are somewhat hard to distinct. And then, for the *seq* and *N_IN_Conn_P_SrcIP* the distribution by attack are almost identical between these 2 features, but also overlapped for each one of them within the typical range of their change. Given the almost 59 times difference in the χ^2 parameter between the most informative *mean* feature and *N_IN_Conn_P_DstIP*, it is reasonable to try a classification without that feature and also the *seq* feature, since their distributions are very similar. In other words, in our experimentation we perform classification tests once with the full set of 10 features, and a second time – with only 8 features in order to evaluate the efficiency of this reduction by classification accuracy and execution time.



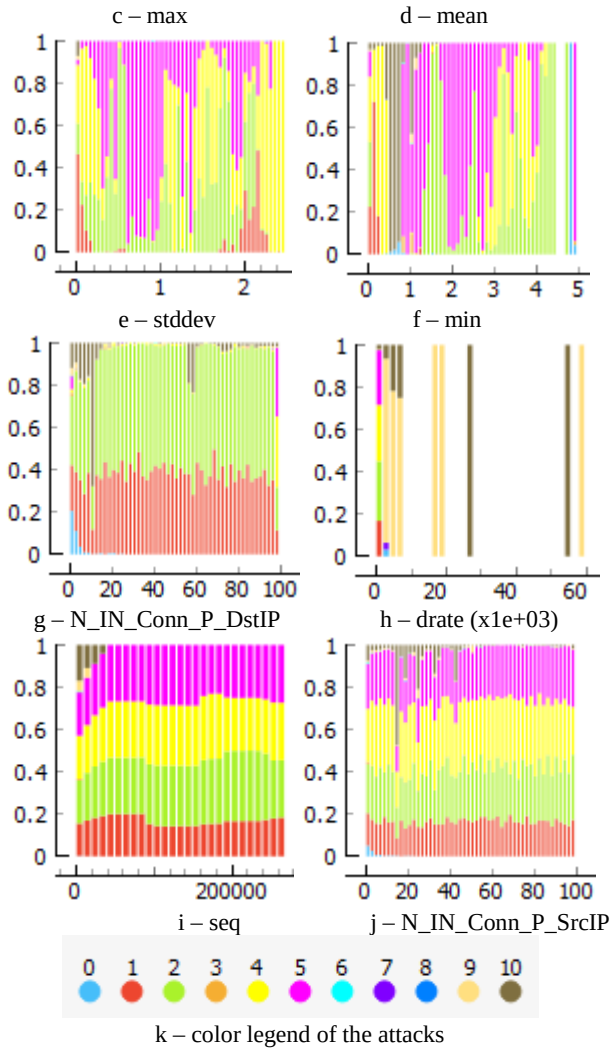


Fig. 1: Probability of different kinds of attacks, given particular value of a feature

2.2 SVM Description

SVM is supervised training algorithm [12], which for the purposes of the current study produce a model, capable of discriminating samples from malicious network activities from those of the normal traffic, and also as a separate model – classifying various types of DDoS attacks. In the first case, which could also be viewed as binary classification problem, the following simplified graphical representation (Fig. 2) may be used.

A hyperplane needs to be found, which will separate in the most discriminating fashion the clusters of samples, corresponding to the different classes, in this instance the attack samples and those, calculated from the normal traffic (non-attack samples). All samples, which are located in minimal distance to the hyperplane, are known as support vectors, while the distance itself is called a margin. The very idea of the algorithm is to find a hyperplane, which maximizes the margin.

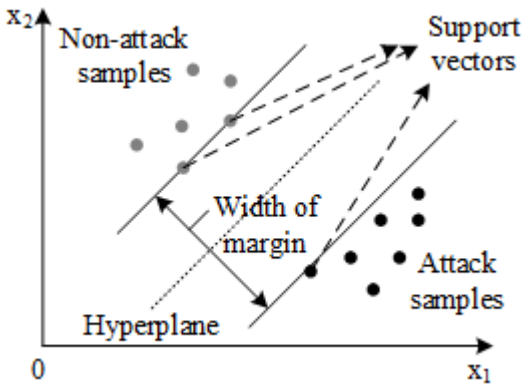


Fig 2: SVM operational principle

In *N* dimensions, the hyperplane equation could be expressed as [12]:

$$y = w_0 + w_1x_1 + w_2x_2 + \cdots + w_Nx_N = w_0 + \sum_{i=1}^{N} w_ix_i = w_0 + \boldsymbol{w}^T\boldsymbol{X} = b + \boldsymbol{w}^T\boldsymbol{X}, \quad (4)$$

where $\boldsymbol{w}_i = (w_{0i}, w_{1i}, \ldots, w_{Ni})^T$ is a vector of weights, b is a bias, equal to w_0, and $\boldsymbol{X}_i = (x_{1i}, x_{2i}, \ldots, x_{Ni})$ is a vector, representing an input sample to be discriminated from the samples, belonging to the other cluster. If the following condition is met [12]:

$$y_i(\boldsymbol{w}^T\boldsymbol{X}_i + b) \geq 1, \quad (5)$$

then $\boldsymbol{X}_i$ would be associated with the correct class. If all vector points from both classes are linearly separable, then the hyperplane, satisfying the above relations will completely separate the classes, but if a new point comes to a class, and falls on the other side of the hyperplane – it will be incorrectly classified. This is known as SVM with *hard margin*.

In order to overcome the limitations of the strict rule from above, a slack variable ξ is put in (5) [12]:

$$y_i(\boldsymbol{w}^T\boldsymbol{X}_i + b) \geq 1 - \xi_i, \quad (6)$$

and there will be correct classification only when $\xi_i = 0$. For every case $\xi_i > 0$, ξ_i represents the error of classification, which in average after all classifications will be [12]:

$$\bar{\xi} = \frac{1}{n}\sum_{i=1}^{n}\xi_i. \quad (7)$$

Then, naturally the following objective function emerges [12]:

$$\min_{w,b}\frac{1}{2}\|w\|^2 + \sum_{i=1}^{n}\xi_i, \quad (8)$$

which must hold true, when (6) is satisfied for all i = 1, 2, …, n, and all input samples are correctly classified. This is known as SVM with *soft margin*.
The loss function is zero, given $Z_i = y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{i}} + b) \geq 1$, and increasing with the ever stronger condition $Z_i < 1$ [12]. So the loss could be derived from max(0, 1-Z_i).

Using Lagrange multiplier it becomes possible project data from low-dimensional space to higher number of dimensions in order to get better separability of samples from different classes, which is known as the SVM dual form [12]:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j\, y_i y_j \left(\boldsymbol{X}_i^T \boldsymbol{X}_j\right), \quad (9)$$

which should be satisfied, given $\alpha_i \geq 0$ for i = 1, 2, …, n, while $\sum_{i=1}^{n} \alpha_i y_i = 0$. Every $\boldsymbol{X}i$ represents a support vector when $\alpha_i \geq 0$ and not being such, given α_i = 0. The dependence of finding the solution for the dual form of SVM falls over α, since intermediate results depend on the scalar product of vector pairs, including the bias b. An ease into the process of finding the dot products is the introduction of kernel and perform all calculations in another space, with higher number of dimensions than the initial one [12]:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j\, K\left(\boldsymbol{X}_i^T . \boldsymbol{X}_j\right) \quad (10)$$

for $0 \leq \alpha_i \leq C$, when i = 1, 2, …, n and $\sum_{i=1}^{n} \alpha_i y_i = 0$.
In our study we use 4 different kernel functions. A linear one, defined by [12]:

$$K(\boldsymbol{X}_1, \boldsymbol{X}_2) = \boldsymbol{X}_1^T \boldsymbol{X}_2, \quad (11)$$

where K is the kernel function, and $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are input vectors.
A polynomial kernel could be expressed as [12]:

$$K(\boldsymbol{X}_1, \boldsymbol{X}_2) = (c + g\boldsymbol{X}_1^T \boldsymbol{X}_2)^d, \quad (12)$$

where d – is degree of the kernel, c – a constant, and g – coefficient of proportion (gamma).
A Gaussian Radial Basis Function (RBF) as a kernel is another option, which could be represented as [12]:

$$K(\boldsymbol{X}_1, \boldsymbol{X}_2) = exp(-g|\boldsymbol{X}_1 - \boldsymbol{X}_2|^2), \quad (13)$$

where $|\boldsymbol{X}_1 - \boldsymbol{X}_2|$ is the Euclidean distance between the vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. With the change of g from small to large values the general observed effect is that the classifier goes from underfitting to overfitting, passing through some optimal configuration
The sigmoidal function as a kernel corresponds to the following equation [12]:

$$K(\boldsymbol{X}_1, \boldsymbol{X}_2) = \tanh(g\boldsymbol{X}_1^T \boldsymbol{X}_2 + c). \quad (14)$$

2.3 SVM Optimization Procedure

During experimental testing the following evaluation parameters are found from both the validation process over the full training set and classification over the full test set:

- *AUC* – the Area under ROC – the Receiver Operating Curve;
- *CA* – Classification Accuracy – the ratio of correctly marked samples with regard to all input samples;
- *Precision* – the part of the actual truly classified instances among all marked as positive instances;
- *Recall* – the part of the actual truly classified instances with regard to the whole number of positive instances in the dataset of the same type (class);
- *F1* – harmonic mean, taking as input the precision and recall parameters;
- *Specificity* – the part of the actual marked true negatives, related to all negative samples from the input;
- *LogLoss* – it is the loss value, for which the cross-entropy function is used, and it accounts for the uncertainty level of the prediction being made (to which class a test sample belongs); it depends on the variation degree from the actual class.
- *Train Time* – the full time needed to train the classifier;
- *Test Time* – the full time, necessary to make classification over the testset after the classifier has been completely trained.
- *Confusion Matrix* – a square matrix representation of the predicted samples by class (as rows) and the actual class of each sample (as columns).

The general optimization procedure that we propose here is presented in Fig. 3. According to the recommendations, given in [13], the initial value for the Cost parameter of the SVM is C = 1, the constant term c = 1, the degree d = 3, the Numerical Tolerance NT = 10^{-3}, the Iteration Limit IL = 10^5. One of the major effects from the training is finding the optimal value for g. As empirical rule [13], the initial value for it may be set to 1/k, where k is the number of components of the feature vectors, that is 0.1 in our case. The following categorical variable is introduced, denoting the type of the SVM kernel – t = 1 for Linear, t = 2 – for Polynomial, t = 3 – for

RBF, and t = 4 – for Sigmoid. The current iteration during training is denoted with *il*.

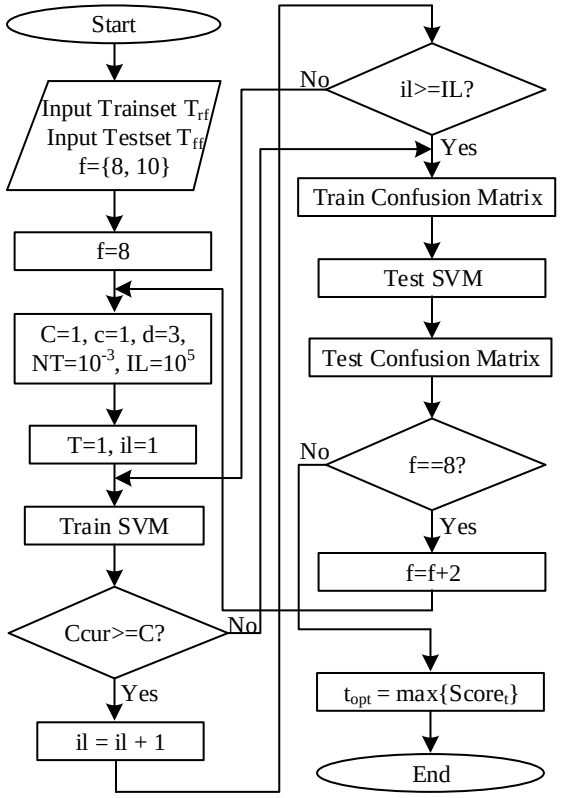


Fig. 3: Optimization procedure for the SVM

3 Experimental Results

Experimental results are gathered using the IBM PC compatible computer with Intel Xeon E5-1620 processor, comprising of 4 cores operating in hyper threading mode at 3.5 GHz. The amount of cache at first level is 256 kB, 1 MB – at the second and 10 MB – at the third one, while the size of the RAM is 64 GB. Testing is being performed within the Orange v. 3.28 application for machine learning under the control of Microsoft Windows 10 Professional operating system.

Processing times for the training and testing phase of the SVM, which performs simple discrimination of the traffic to normal (Class 0) and attack (Class 1), are given in Table 2.

The time periods from Table 2 correspond to maximum number of 100 000 training iterations, set in advance as a limit. In an additional experiment SVM for binary discrimination is tested at 1 000 000 maximum iterations only for the RBF kernel. It turns out that the training time is 62 620 s, the validation time – 115 s, and the testing time 31 s, using 10 features. Obviously, the criteria for terminating the learning process is met long before reaching the iterations number limit, almost equal to 100 000 iterations.

Table 2. SVM processing times for detecting malicious network activities vs. normal traffic

SVM type	Features number	Training time, s	Validation time, s	Testing time, s
Linear	8	8 776	68	15
	10	11 760	134	34
Polynomial	8	21 341	85	26
	10	12 677	143	36
RBF	8	76 539	228	57
	10	71 458	272	70
Sigmoid	8	137 980	218	54
	10	23 425	145	32

The processing times when classifying multiple attacks with a set limit of 1 000 000 iterations for the SVM, using RBF, which turns out to be the best option among the tested kernels, as the experimental results from below demonstrate, are given in Table 3.

Table 3: SVM processing times for classifying various attacks

SVM type	Features number	Training time, s	Validation time, s	Testing time, s
RBF	8	1124612	55 543	32 558
	10	508 111	133 136	33 237

The detection rate of attacks vs. non-attacks, measured both during the testing and training phase are given in Table 4.

Table 4. Detected attacks as proportion of the actual attacks

SVM type	Features number	Non-attacks, %		Attacks, %	
		Train	Test	Train	Test
Linear	8	2.7	3.7	100.0	100.0
	10	2.7	3.7	100.0	100.0
Polyno-mial	8	10.5	8.4	100.0	100.0
	10	33.8	29.9	100.0	100.0
RBF	8	11.9	11.2	100.0	100.0
	10	35.7	33.6	100.0	100.0
Sigmoid	8	0.8	0.9	100.0	100.0
	10	2.2	1.9	100.0	100.0

All evaluation parameters of the detection efficiency, using all 10 features on the train set as a full validation for the 4 kernels (Fcn.) of the SVM, are visible in Table 5. Class (Cl.) 0 corresponds to non-attack, and 1 – to attack.

Table 5. Attack detection efficiency on the train set, using 10 features

Cl.	Fnc	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	Lin.	0.9866	0.9998	0.0526	1.0	0.0270	0.0010	1.0
	Pol.	0.9977	0.9999	0.4882	0.8802	0.3378	0.0020	0.9999
	Rbf	0.8322	0.9999	0.5217	0.9705	0.3567	0.0010	0.9999
	Sig.	0.6936	0.9998	0.0295	0.0465	0.0216	0.0022	0.9999
1	Lin.	0.9866	0.9998	0.9999	0.9998	1.0	0.0010	0.0270
	Pol.	0.9977	0.9999	0.9999	0.9999	0.9999	0.0020	0.3378
	Rbf	0.8322	0.9999	0.9999	0.9999	0.9999	0.0010	0.3567
	Sig.	0.6936	0.9998	0.9999	0.9998	0.9999	0.0022	0.0216

The detection efficiency results for 10 features when working with the test set are given in Table 6.

Table 6. Attack detection efficiency on the test set, using 10 features

Cl.	Fnc	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	Lin.	0.9809	0.9998	0.0720	1.0	0.0373	0.0017	1.0
	Pol.	0.9972	0.9998	0.4475	0.8888	0.2990	0.0024	0.9999
	Rbf	0.8121	0.9999	0.5	0.9729	0.3364	0.0013	0.9999
	Sig.	0.6973	0.9997	0.0246	0.0363	0.0186	0.0024	0.9999
1	Lin.	0.9809	0.9998	0.9999	0.9998	1.0	0.0017	0.0373
	Pol.	0.9972	0.9998	0.9999	0.9998	0.9999	0.0024	0.2990
	Rbf	0.8121	0.9999	0.9999	0.9999	0.9999	0.0013	0.3364
	Sig.	0.6973	0.9997	0.9998	0.9998	0.9999	0.0024	0.0186

Complete validation over the train set for the detection capabilities of the SVM, using 8 features, leads to the results from Table 7.

Table 7. Attack detection efficiency on the train set, using 8 features

Cl.	Fnc	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	Lin.	0.9846	0.9998	0.0526	1.0	0.0270	0.0017	1.0
	Pol.	0.9250	0.9998	0.1884	0.8863	0.1054	0.0021	0.9999
	Rbf	0.7235	0.9998	0.2110	0.9361	0.1189	0.0011	0.9999
	Sig.	0.6076	0.9997	0.0097	0.0122	0.0081	0.0020	0.9999
1	Lin.	0.9846	0.9998	0.9999	0.9998	1.0	0.0017	0.0270
	Pol.	0.9250	0.9998	0.9999	0.9998	0.9999	0.0021	0.1054
	Rbf	0.7235	0.9998	0.9999	0.9998	0.9999	0.0011	0.1189
	Sig.	0.6076	0.9997	0.9998	0.9998	0.9999	0.0020	0.0081

The unknown samples from the test set cause the detection results, again for 8 features, shown in Table 8.

Table 8. Attack detection efficiency on the test set, using 8 features

Cl.	Fnc	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	Lin.	0.9781	0.9998	0.0720	1.0	0.0373	0.0019	1.0
	Pol.	0.9157	0.9998	0.1525	0.8181	0.0841	0.0024	0.9999
	Rbf	0.6884	0.9998	0.2016	1.0	0.1121	0.0014	1.0
	Sig.	0.6109	0.9997	0.0107	0.0126	0.0093	0.0023	0.9998
1	Lin.	0.9781	0.9998	0.9999	0.9998	1.0	0.0019	0.0373
	Pol.	0.9157	0.9998	0.9999	0.9998	0.9999	0.0024	0.0841
	Rbf	0.6884	0.9998	0.9999	0.9998	1.0	0.0014	0.1121
	Sig.	0.6109	0.9997	0.9998	0.9998	0.9998	0.0023	0.0093

When working with the SVM as a classifier of multiple attacks, using only the RBF kernel, the relative amount of correctly discovered attack instances is given in Table 9.

Table 9. Correctly classified attacks as portion of the actual attacks in %

Attack	Train set 8 feats.	Test set 8 feats.	Train set 10 feats.	Test set 10 feats.
0	27.0	26.2	66.5	66.4
1	71.6	71.7	96.7	96.6
2	98.9	98.8	99.0	99.0
3	8.8	10.0	25.8	26.6
4	94.0	93.9	91.7	91.7
5	95.0	95.0	97.1	97.1
6	0.0	0.0	4.2	2.0
7	22.0	35.7	18.6	35.7
8	0.0	0.0	0.0	0.0
9	4.7	5.1	14.6	15.5
10	96.3	96.2	95.5	95.6

The RBF kernel is used for that purpose, because it yields the highest detection rate with comparison to the other 3 types of kernels, as shown in Tables 4-8.

The confusion matrices from classifying attacks over the test set at 10 and 8 features are depicted in Fig. 4.

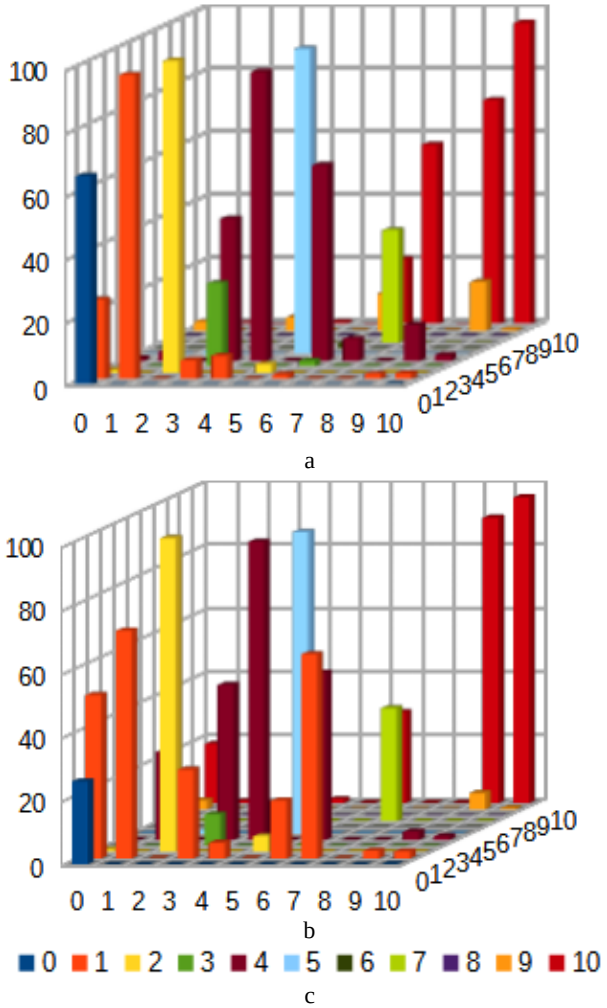


Fig. 4: Confusion matrix from classification if the test set at a – 10 features, b – 8 features, c- color legend of the classes

Classification efficiency at 10 features is shown in Table 10 for the train set.

Table 10. Classification efficiency of SVM over all types of attacks at 10 features after full validation over the training set

Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	0.9992	0.9999	0.6852	0.7068	0.6648	0.0003	0.9999
1	0.9949	0.9739	0.9255	0.8879	0.9666	0.0634	0.9753
2	0.9988	0.9895	0.9816	0.9735	0.9898	0.0362	0.9894
3	0.9993	0.9996	0.4002	0.8970	0.2576	0.0009	0.9999
4	0.9959	0.9716	0.9451	0.9747	0.9173	0.0690	0.9913
5	0.9981	0.9895	0.9796	0.9887	0.9706	0.0375	0.9961
6	0.9994	0.9997	0.0805	1.0	0.0419	0.0006	1.0
7	0.9999	0.9999	0.2972	0.7333	0.1864	4.66e-5	0.9999
8	0.9999	0.9999	N/A	N/A	N/A	2.45e-5	1.0
9	0.9942	0.9954	0.2381	0.6502	0.1458	0.0115	0.9996
10	0.9986	0.9930	0.8456	0.7585	0.9553	0.0142	0.9938

Applying the test set as input to the RBF SVM classifier at 10 features, we get the results from Table 11.

Table 11. Classification efficiency of SVM over all types of attacks at 10 features after processing the test set

Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	0.9992	0.9999	0.6826	0.7029	0.6635	0.0003	0.9999
1	0.9949	0.9737	0.9252	0.8876	0.9661	0.0637	0.9753
2	0.9988	0.9895	0.9816	0.9736	0.9898	0.0363	0.9894
3	0.9975	0.9996	0.4123	0.9195	0.2657	0.0009	0.9999
4	0.9958	0.9715	0.9448	0.9744	0.9170	0.0692	0.9912
5	0.9981	0.9895	0.9797	0.9887	0.9708	0.0375	0.9961
6	0.9994	0.9997	0.0386	1.0	0.0197	0.0006	1.0
7	0.9999	0.9999	0.5263	1.0	0.3571	3.84e-5	1.0
8	0.9936	0.9954	0.2511	0.6631	0.1549	0.0115	0.9996
9	0.9987	0.9930	0.8458	0.7582	0.9564	0.0140	0.9938
10	0.9986	0.9930	0.8456	0.7585	0.9553	0.0142	0.9938

Using just only the 8 features cause the RBF SVM to produce a classification result over the train set, as shown in Table 12.

Table 12. Classification efficiency of SVM over all types of attacks at 8 features after full validation over the training set

Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	0.9853	0.9998	0.3816	0.6493	0.2702	0.0005	0.9999
1	0.9852	0.9386	0.7966	0.8971	0.7163	0.1269	0.9834
2	0.9970	0.9839	0.9719	0.9558	0.9885	0.0564	0.9820
3	0.9677	0.9996	0.1569	0.7375	0.0878	0.0021	0.9999
4	0.9887	0.9366	0.8877	0.8412	0.9395	0.1316	0.9355
5	0.9965	0.9839	0.9682	0.9870	0.9502	0.0556	0.9956
6	0.9523	0.9997	N/A	N/A	N/A	0.0014	1.0
7	0.9884	0.9999	0.3421	0.7647	0.2203	0.0001	0.9999
8	0.9909	0.9999	N/A	N/A	N/A	5.43e-5	1.0
9	0.9882	0.9952	0.0894	0.7784	0.0474	0.0138	0.9999
10	0.9960	0.9905	0.8022	0.6874	0.9630	0.0173	0.9910

The test set in the same time at 8 features has been classified to an extent, represented by the parameters from Table 13.

Table 13. Classification efficiency of SVM over all types of attacks at 8 features after processing the test set

Cl.	AUC	CA	F1	Pre-cision	Recall	Log-loss	Specifi-city
0	0.9857	0.9998	0.3708	0.6363	0.2616	0.0006	0.9999
1	0.9851	0.9385	0.7966	0.8964	0.7167	0.1271	0.9832
2	0.9968	0.9838	0.9718	0.9558	0.9883	0.0574	0.9821
3	0.9707	0.9996	0.1780	0.8333	0.0996	0.0020	0.9999
4	0.9886	0.9365	0.8873	0.8410	0.9391	0.1318	0.9356
5	0.9965	0.9838	0.9682	0.9868	0.9503	0.0562	0.9955
6	0.9512	0.9997	N/A	N/A	N/A	0.0015	1.0
7	0.9878	0.9999	0.5263	1.0	0.3571	0.0001	1.0
8	0.9882	0.9952	0.0950	0.7922	0.0505	0.0138	0.9999
9	0.9960	0.9905	0.8008	0.6858	0.9622	0.0173	0.9910
10	0.9960	0.9905	0.8022	0.6874	0.9630	0.0173	0.9910

4 Discussion

The fastest SVM detector of DDoS attacks in terms of learning time is the one, using a Linear kernel, followed by that with Polynomial kernel, 1.08 times at 10 features (Table 2). Then follows the classifier with Sigmoid kernel, around 2 times slower than the linear one and the last is the implementation with the RBF kernel – around 6 times slower. It is worth noting that for all detectors, apart the linear one, the training time at 8 features is longer than that at 10 features – the reduction of the information, included in the training set affects the number of iterations until reaching the target value of the cost function or until using the whole designated learning period value. Similar are the relations of the time periods during the validation phase. The RBF based SVM is the slowest one with almost 2 times longer classification process than the fastest Linear SVM. This tendency is preserved at the testing phase as well. With the exception of the Sigmoid SVM, for all other implementations it takes less time to classify at 8 features, than at 10, e.g. more than twice a difference for the Linear SVM.

The most accurate SVM detector of DDoS attacks is the one with RBF kernel, achieving 35.7 % at 10 features detection rate for the non-attack samples (Table 4) during validation and 33.6% during testing. More than 3 times lower is the detection rate for the same samples when applying only 8 features. All attack samples in the same time are being completely correctly discriminated during both phases. The latter is also true for the rest of the detectors with the Sigmoid SVM detecting just 0.8 % of the attack samples during training and 0.9% - during testing. Only the Linear SVM yields identical detection rate when operating over 8 or 10 features, which means that if is being applied in practice, it

would be preferable to work with 8 features, due to the twice faster classification. These results led to the conclusion that further testing with classifiers, capable of discriminating all 10 types of attacks should be done with a SVM, using RBF kernel. All observations from above are also supported by the evaluating parameters from Tables 5-8.

Classifying all 10 DDoS attacks (indexes from 1 to 10) and recognizing non-attack samples in the same time (index 0) took the RBF SVM almost twice longer, using 8 features rather than 10, during training (Table 3). In the same time, 8 feature lead to more than twice faster validation with comparison to the 10 feature implementation. The most accurately spotted attack is DoS UDP flood (Table 9), followed by the DDoS UDP flood, DoS TCP flood, Service Scan, and so on. The most hard to discover attacks are the Data Exfiltration and DDoS HTTP flood. The most probable reason for this is the considerably smaller number of samples for these attacks, present in the training set, compared to the amount of samples for the rest types of attacks. Nevertheless, the proportion of data exchanged during the various tested attacks corresponds to real-world scenarios and the observed dependency should be taken as inherited peculiarity of the single SVM classifier itself. Obviously, to get as close as possible detection rate for these rarely spot types of attacks, one possible direction for future work it would be to construct a cascade of classifiers. The variation between the number of discovered attacks between the phases of training and testing is negligible. When using 8 features, differences in detection accuracy for some of the attacks, compared to that for 10 features, goes as high as 3 times, as it is in the case of OS Fingerprint, or around 40% for the non-attack samples (Table 9).

The most mismatched non-attack samples, using 10 features (Fig. 4 a), are recognized as DoS TCP flood (25.2%), the DoS TCP attack – with DDoS TCP flood (3.0%), DoS UDP flood – with DDoS UDP flood (1.0%), DoS HTTP flood – with DDoS TCP flood (45.2%), which is with 81.6% higher than the correctly found samples, DDoS TCP flood – with DoS TCP flood (7.5%), DDoS UDP flood – with DoS UDP flood (2.9%), DDoS HTTP flood – with DDoS TCP flood (62.1%), close to 60% higher than the number of the correctly recognized samples for this particular attack, Keylogging – with Service Scan (57.1%), OS Fingerprint – with Service Scan (71.2%), again serious mismatch rate, and Service Scan – with DDoS TCP (2.0%). All this ratios could be observed from the confusion matrix after classification over the test set, representing the proportion of the classified samples by attack from the actual number of samples for the same attack, as shown in Fig. 4 – for 10 features in Fig. 4 a and for 8 features – in Fig. 4. b. Using 8 features, lead to increase of the proportion of mismatches with closes incorrect type of attack, as follows: twice for non-attack samples, 9 times for the DoS TCP flood, 1.2 times for the DoS UDP flood, 1.08 times for the DoS HTTP flood, 1.5 times decrease for the DDoS TCP flood, 1.7 times for the DDoS UDP flood, 1.2 times decrease for the DDoS HTTP flood, 57.1% decrease for the Keylogging, and 1.7 times decrease – for the Service Scan (Fig. 4 b). Apart from worsening of the classification accuracy for some of the attacks, such as the DoS TCP flood or the non-attack samples, there is also a positive trend for other types of attacks, such as the Keylogging. Decrease of the information redundancy in the training set at 8 features, compared to 10, obviously preserves better some of the relations for attacks, which have smaller intensity as per the exchanged data over the network, such as the Keylogging. It would be practical to use this feature set, although considerably more inaccurate for attacks with high intensity of the generated traffic, for some more rare activities, when specifically searching for them in a monitored network. All these results are also supported by the evaluating parameters, shown in Tables 10-13. For some of the attacks with really small number of instances in the training and the testing set, some of the parameters are hard to calculate, as the denominator of the equations for them, tends to be very small, almost equal to 0, so they are marked I the tables with N/A.

At the end of the discussion section, we make a comparison with another implementation of a binary SVM classifier (detector) of DDoS attacks, proposed by other authors in [11]. It is tested over the same dataset with the same 10 features as in this study and it has the cost parameter being put to C = 1, using a Linear kernel, and having a training time limit of 100 000 iterations. The confusion matrices for this classifier and the best of our binary SVM classifiers (10 features, RBF kernel, 100 000 iterations limit, C = 1) are shown in Table 14 as proportion of the detected samples to all actual of that type ones, given in %.

Table 14. Confusion matrices of compared SVM detectors

Ours, in %			Proposed in [11], in %		
Attack	0	1	Attack	0	1
0	33.6	66.7	0	100.0	0
1	0.0	100.0	1	11.63	88.37

One of the advantages of the SVM detector, proposed in [11], is the higher detection rate of non-attack samples, practically 100%. On the other hand, our implementation achieves 100% correct identification of all attacks, while the SVM from [11] achieves 88.37% correct classifications. Other evaluating parameters, denoting the detection efficiency of both classifiers, are presented in Table 15.

Table 15. Evaluation parameters of compared SVM detectors

Parameter	Ours	Proposed in [11]
Accuracy	0.9999	0.8837
Precision	0.9998	1
Recall	0.9999	0.8837
F1-measure	0.9998	0

Taken on average among both classes – 0 (non-attack) and 1 (attack), given also the size of the test set of close to a million samples, our implementation of a SVM detector show better performance with the exception of the Precision parameter, which is 0.0002 smaller, but this is a difference, which makes both classifiers by this criterion relatively equal. Nonetheless, more work is needed to make the RBF SVM detector more efficient in terms of detecting non-attack samples better. One of the directions for future work is to try a combined type of a classifier, which could achieve better accuracy. Another possible path for further research is the introduction of a sampling, which will increase the number of non-attack samples artificially – an interpolation, which will lead to equalization of the number of normal traffic samples and those from DDoS attacks (floods), which always prevail the normal ones and create a disbalance, into which the non-attack samples is harder to find.

5 Conclusion

In this paper two types of SVM classifiers for DDoS attacks are presented – binary and multiclass ones, covering 10 type of the most popular malicious activities, carried out often with the help of IoT botnets. The most accurate detector is using a RBF kernel, which is also thought as the most proper solution for the multiclass classification. The fastest SVM classifier is the Linear one, followed by the Polynomial, the Sigmoid and the RBF at the end, which holds true for both most of the cases of training and testing. Training with 8 features turns out slower in most of the cases than with 10 features, but testing may be significantly faster with 8 features for some of the kernels, used in SVM. In the multiclass classification, the 10-feature set leads to observable higher accuracy, than the 8-feature set. This effect is less expressed in the binary classification, but still holds true as a general trend. Given achieved accuracies, both the binary and multiclass SVMs, presented in this study in thier optimal configurations, are thought to be applicable in real-world monitoring systems against DDoS attacks. Still, more work is needed, especially into increasing the accuracy of the binary SVM in to discovering non-attack samples over the background of ongoing DDoS attack with its characteristic high intensity. Various strategies could be applied, which include cascade of classifiers, intelligent sampling of the training set and others. All these will be considered during the future work on the problem.

References:

[1] Behal, S., Kumar, K. Trends in Validation of DDoS Research. Procedia Computer Science, Vol. 85, 2016, pp. 7-15.

[2] Ye, J., Cheng, X., Zhu, J., Feng, L., Song, L. A. DDoS Attack Detection Method based on SVM in Software Defined Network. *Security and Communication Networks*, Vol. 2018, 2018, Article ID 9804061.

[3] Yusof, A. R. A., Udzir, N. I., Selamat, A. An Evaluation on KNN-SVM Algorithm for Detection and Prediction of DDoS Attack. *In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, Cham, August 2016, pp. 95-102.

[4] Daneshgadeh, S., Kemmerich, T., Ahmed, T., Baykal, N. An Empirical Investigation of DDoS and Flash Event Detection using Shannon Entropy, KOAD and SVM combined. *In 2019 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2019, pp. 658-662

[5] Khuphiran, P., Leelaprute, P., Uthayopas, P., Ichikawa, K., Watanakeesuntorn, W. Performance Comparison of Machine Learning Models for DDoS Attacks Detection. *In 2018 22nd International Computer Science and Engineering Conference (ICSEC)*, IEEE, , November 2018, pp. 1-4.

[6] Ali, J., Roh, B. H., Lee, B., Oh, J., Adil, M. A Machine Learning Framework for Prevention of Software-Defined Networking controller from DDoS Attacks and Dimensionality Reduction of Big Data. *In 2020 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, October 2020, pp. 515-519.

[7] Adhikary, K., Bhushan, S., Kumar, S., Dutta, K. Hybrid Algorithm to Detect DDoS Attacks in VANETs. *Wireless Personal Communications*, Vol. 114, No. 4, 2020, pp. 3613-3634.

[8] Nazih, W., Hifny, Y., Elkilani, W., Abdelkader, T., Faheem, H. Efficient Detection of Attacks in SIP based VoIP Networks using Linear l_1-SVM Classifier. *International Journal of Computers Communications & Control*, Vol. 14, No. 4, 2019, pp. 518-529.

[9] Kajal, A., Nandal, S. K. ABC-ANN-SVM Hybrid Approach to Enhance Cyber Security against Malware, DDoS Attacks. *Journal of Critical Reviews*, Vol. 7, Issue 19, 2020, pp. 4557-4570.

[10] Arshi, M., Nasreen, M. D., Madhavi, K. A Survey of DDOS Attacks using Machine Learning Techniques. *In E3S Web of Conferences*, Vol. 184, 2020, p. 01052.

[11] 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.

[12] Wilmott, P., *Machine Learning: An Applied Mathematics Introduction*, Panda Ohana Publishing, 2019.

[13] SVM, Orang Visual Programming, Orange Data Mining, https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/svm.html, last accessed on August 4th, 2021.

[14] Nikulin, M. S., Chimitova, E. V. *Chi-squared Goodness-of-fit Tests for Censored Data*, Wiley, 2017.