Machine Learning Detection of DDoS Attacks Based on Visualization of Recurrence Plots

Lyudmyla Kirichenko[1][0000-0002-2780-7993], Petro Zinchenko[1], Tamara Radivilova[1][0000-0001-5975-0269], Maksym Tavalbeh[1]

[1] Kharkiv National University of Radio Electronics, Kharkiv, 14 ave.Nauki, Ukraine
tamara.radivilova@gmail.com

Abstract. The article considers a new method of detecting DDoS attacks based on the construction of recurrence plots. Time traffic realization is transformed into a matrix consisting of 0 and 1, which characterizes the recurrence of the realization. The matrix is presented in the form of a black-and-white image. Then the residual neural network is used to classify images. This approach is used to detect the realizations of the attacked network traffic. The results showed that the proposed method has a sufficiently high accuracy of classification and can be used to detect attacks in intrusion detection systems.

Keywords: fractal time series, machine learning classification, recurrence plot, DDoS-attack, data traffic, deep residual networks

1 Introduction

Attack detection is the process of monitoring events occurring in a computer system to detect signs of possible unauthorized access to the network. Time series models play an important role in the study of system behavior models for detecting attacks such as Distributed Denial of Service (DDoS), Detecting Account Takeover (ATO), Data Exfiltration [1-4]. They are successfully used for classification and forecasting, which are performed by methods of machine learning [5-7].

Inbound traffic modeling uses flow-related data at layers 4 and 7 of the OSI network model, which is analogous to DDoS attacks. The resulting model attacks are superimposed on real untreated traffic to determine the key points of intrusion warning creation [8,9].

In recent years, machine learning methods have been used to analyze and classify time series. One of the solutions to the problem of timely attack detection is to develop a classifier based on machine learning, which would determine whether the incoming traffic is under attack.

The task of time series classification, which includes the attacked traffic detection, is one of the most complex tasks of machine learning. Several approaches to time series classification are known, most of which are based on calculation of statistical characteristics of time series or calculation of different metrics between time series [10,11].

In recent years, a number of works have appeared in which the approach on the basis of the recurrence analysis, which was originally proposed in [12] and further developed in [13, 14], is used to classify time series by machine learning methods. With the development of machine learning methods, the characteristics obtained from the recurrence plots were used as characteristics for classification tasks [15, 16].

The recurrence plot method is based on the repeatability of time series states (recurrence). Recurrence properties of time series are represented in the form of a matrix with a certain geometric structure, which can be visualized and classified by images of recurrence plots [17,18].

The purpose of this article is a comparative analysis of the DDoS attacks detection with the help of machine learning methods, based on the visualization of recurrence plots.

2 Time series classification using recurrence plots

The recurrence plot method is based on the fact that by building a phase trajectory based on only one variable of the system it is possible to restore the topology of the full phase trajectory, as in the case of all system components. This approach was proposed in [19]. The method of attractor reconstruction by one time realization (Packard-Tackens procedure) is the basis of almost all time series analysis algorithms by nonlinear dynamics methods:

$$x_i = \left(u_i, u_{i+\tau}, ..., u_{i+(\mathrm{m}-1)\tau}\right), \tag{1}$$

where x_i is the value of the phase trajectory x at the moment *i*, u_i is the value of some component of the system *u* at the moment *i*, m is the embedding dimension of the phase space, τ is the time delay.

In [12] a way of displaying the m-dimensional phase trajectory of a system's states $\vec{x}(t)$ in length N on a binary matrix of the size $N \times N$ in which the value of 1 corresponds to the repeating of the state at some other time *i* at some other time *j*, and the coordinate axes of the matrix are the time axes was proposed. Such matrix (recurrence plot) records information about recurrence behavior of the system.

Thus, recurrence is defined as a sufficient proximity of the state $\vec{x}_j$ to the state $\vec{x}_i$: recurrent are the states $\vec{x}_j$ that fall into the m-dimensional neighborhood with a radius ε of the point $\vec{x}_i$.

Let the point x_i correspond to the point of the phase trajectory describing the dynamic system represented by the time series $x(t)$ in m -dimensional space at the moment of time $t = i$ for *i = 1,...,N*. Then the recurrence plot *RP* is an array of points where a non-zero element with coordinates (i, j) corresponds to the case when the distance between x_j and x_i is smaller ε:

$$RP_{i,j} = \Theta(\varepsilon - \| x_i - x_j \|), \quad x_i, x_J \in R^m,\ i, j = 1,...N,$$

where ε is the size of the neighborhood of x_i, $\left\| x_i - x_j \right\|$ is the distance between the points, $\Theta(\cdot)$ - the Heaviside function.

Recurrence plots can be represented in black and white. In this case, the recurrence of the state at the moment *i* at different values of time *j* is reproduced inside the two-dimensional square matrix with black and white dots, where black dots indicate the presence of recurrence. Fig. 1 above shows the time realization of a sinusoid (left) and a noisy sinusoid (right). The bottom part of Fig. 1 shows the recurrence plots corresponding to these time realizations.

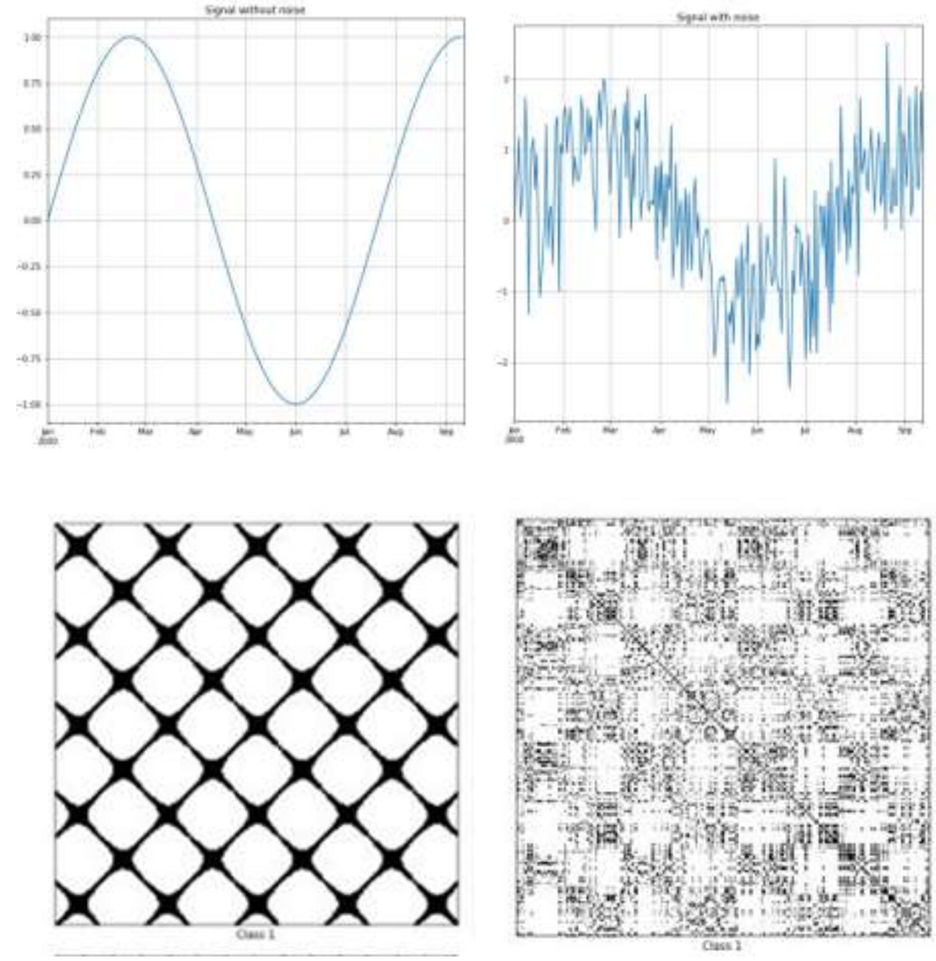

Fig. 1. Time series and corresponding recurrence plots

The analysis of the recurrence plots topology allows to classify observed processes: to define homogeneous processes with independent random values; processes with slowly changing parameters; periodic or oscillating processes corresponding to nonlinear systems, etc. It has been shown in [6, 20] that changes in the correlation structure of the time series cause changes in the topology of the corresponding recurrence plot.

3 Simulation of normal and attacked traffic realizations

An experiment was carried out to detect DDoS attacks, which was based on a recurrence plot to identify the attacked traffic. To train traffic classifier it is necessary to build a training and test sample containing two classes of traffic. In this case, it is advisable to use modeling to realize normal traffic, and to present the attacked traffic as the sum of traffic and the realization of the attack.

It is known that infocommunication traffic has a fractal structure (self-similarity) and long-term dependence. A number of recent studies have shown that under the influence of an attack, traffic changes its self-similar properties and correlation dependencies [21-26]. Therefore, a model based on the self-similarity property was chosen to generate traffic. In work [27] the model of fractal traffic which is generated on the basis of fractal Brownian motion was offered.

Fractal Brownian motion is the most known and simple model of fractal process, which has the only parameter of scaling Hurst index H . The increments of fractal Brownian motion are the Gaussian stationary process with a long-term correlation dependence.

The fractal traffic model is an exponential transformation of a series of fractal Brownian increments

$$Y(t) = \mathrm{Exp}[k * X(t)], \qquad (2)$$

where $X(t)$ is a time series of fractal Brownian motion increments with discrete time with given Hurst index H , k is some coefficient determining the coefficient of series variation $Y(t)$, i.e. the bursts value in realization. Thus, using (2) we obtain the traffic realization with a given degree of long-term dependence determined by the index H and bursts determined by the coefficient k .

Fig.2 shows two model traffic realization with the same Hurst value and different coefficient values k. Both realizations have the same average value, but different degrees of bursts: the upper realization has a maximum burst value of up to 80 and the lower realization has a maximum value of up to 200.

In order to simulate the attacks realizations, the data set described in detail in [28] was used. This work presents the mechanism of collecting real statistical data of SNMP-MIB and their usage. There were conducted real experiments in which there were six types of DoS-attacks and Brute Force attacks. The traffic data were collected from the SNMP agent. The data set consists of 4998 records, where each record consists of 34 MIB variables that are classified into corresponding groups, namely: interface, IP, TCP and ICMP. The figure 3 shows some of the attack realizations used to model the attacked traffic.

Various model attacked traffic realizations can be obtained by summing up the traffic and attack realizations and varying the attack level

$$T_A(t) = T(t) + A(t) * level_A, \tag{3}$$

where $T(t)$ is the traffic realization received in accordance with the transformation (2), $A(t)$ is the attack realization taken from the data set [28], $A(t)$ is the attack level obtained by the ratio of the average value of the attack realization to the traffic realization:

$$level_A = \frac{\overline{A(t)}}{\overline{T(t)}}.$$

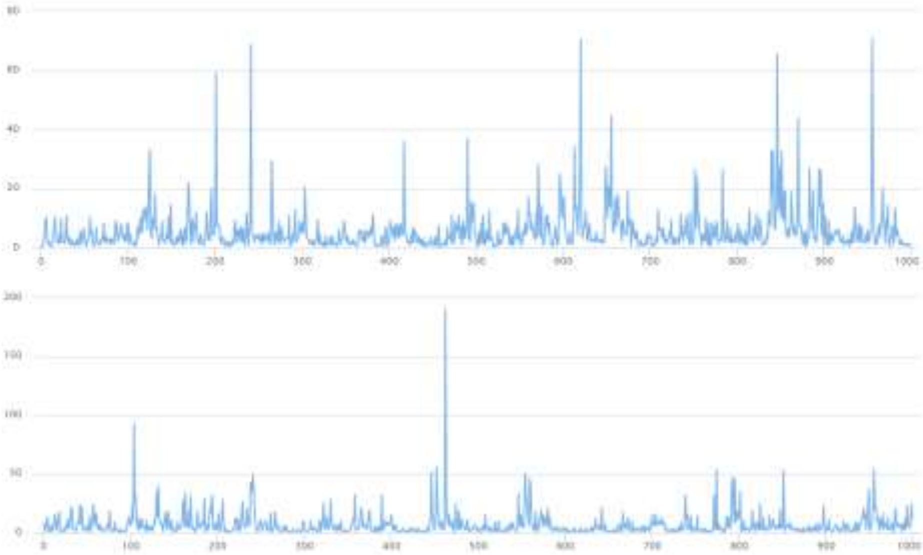

Fig. 2. Model traffic realization of with the same degree of long-term dependence and different degree of emissions

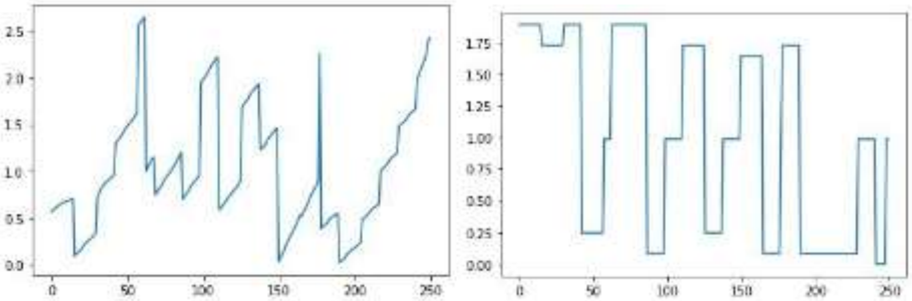

Fig. 3. Some of the attack realizations

Fig. 4 above shows the traffic and attack realization (the attack is shown in a different color) and below the attacked traffic realization with the attack level $level_A = 30\%$ with the length of 250 values.

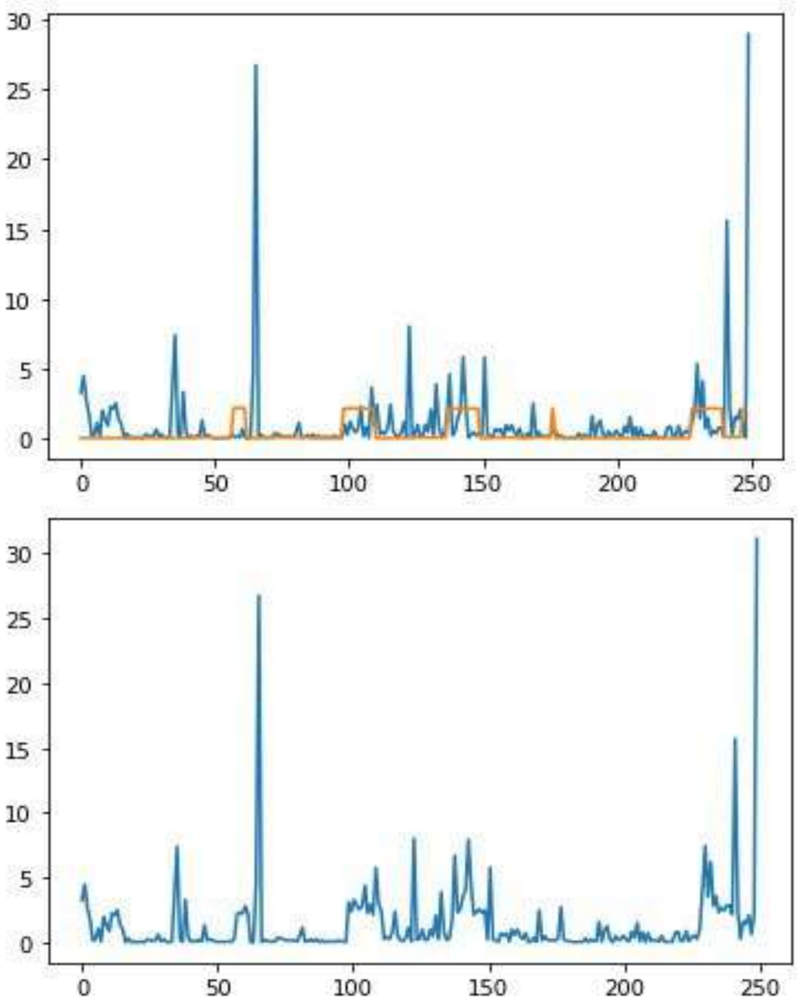

Fig. 4. Top: traffic and attacks realizations (traffic-blue, attack-orange); bottom: traffic realization with 30% attack level

4 Neural networks for image classification

Computer vision technologies are very common. Due to the growth of computing power and the emergence of major image bases it has become possible to use deep neural networks. The best results in the field of image recognition are shown by the Convolutional Neural Network [29, 30]. Their success is conditioned by the possibility of taking into account two-dimensional image topology, as opposed to multilayer perceptron.

Deep convolution neural networks extract low, medium and high level signs through multilayer method. It is logical to assume that if the number of layers is increased, thus increasing the number of features, the quality of recognition will improve. However, studies have shown that this is a problem: as the depth of the network increases, accuracy increases first and then rapidly decreases. One of the reasons for the decline in learning accuracy is the attenuation of the gradient.

Since convolution networks are trained by the method of error back propagation, the error on the last layer is first calculated then on the second-to-last layer and so on. Gradients on the output layer are easily calculated and significant changes in the neuronal weights of the output layer are obtained. For any hidden layer it is necessary to accumulate errors of all following layers. The value of the weight gradient of a deeper layer neuron is proportional to the value of the derivative of the activation function at the point obtained at the direct pass. And as a consequence, the error decreases by about 10 times every 3-4 layers, and if the network is deep enough, the layers that are closer to the beginning are poorly trained or not at all trained.

To overcome this problem, a deep residual network structure has been proposed, the main idea of which is to use a quick-access connection instead of a conventional serial layer connection [31, 32].

Quick access connections skip one or more layers and do the identification matching. Their outputs are added to the outputs of grouped layers. The main network block consists of two layers with weights, not necessarily a convolution and a quick access connection that simply transmits the input signal to the output. The connection diagram is shown in Fig. 5.

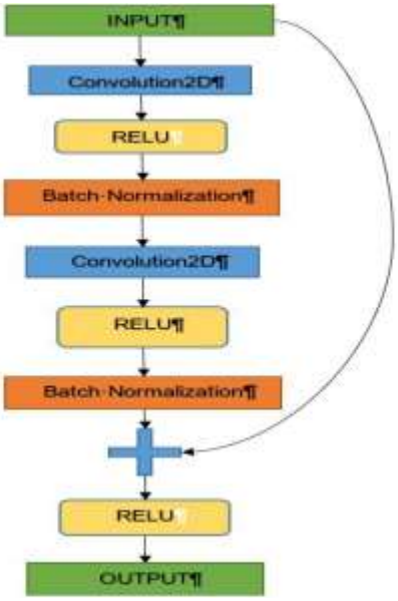

Fig. 5. Layout of the structure of the residual network

5 Experiment description and results

The experiment used a residual neural network consisting of ten blocks, 3 convolution layers and one full-connected layer. The output of the last full-connected layer is input to the logistic function, which performs the distribution into 2 classes. Neurons in the full-connected layer are connected with all neurons in the previous layer. Subsample layers follow the second and third convolution layers. Nonlinearity of ReLU is applied to the output of each convolution and full-connected layer.

The first convolution layer filters the input image with a size of 249×249 pixels and 32 cores with a size of 3×3. The second convolution level takes the output of the first convolution layer as the input data and filters it with 64 cores of 3×3 size. This is followed by a sub-sampling layer. Next come 5 residual blocks with convolutions of 64 cores of 3x3 size. The third convolution layer has 128 cores of 3×3 size, followed by the sub-sampling layer. Next, there are 5 residual blocks with 128 cores of 3x3 size convolutions. The output of the last block is transferred to the global subsample layer. After the sub-sampling layer, a full-connected layer of 256 neurons follows. Adam, an adaptive learning rate optimization algorithm, was used for network learning.

Two samples were generated each time for the classification: a training sample and a test sample. Both samples contained realizations of two classes: attacked and non-attacked traffic. The model traffic was built on the basis of exponential transformation (2). In the case of an attack, one of a DDoS-attack realization from the data set [28] was added to the traffic by formula (3).

At the input of the classifier, recurrence plots calculated on the basis of traffic realizations were applied. The output data were values 1 or 0: presence or absence of DDoS-attack in time series of traffic. Fig.6 shows the recurrence plots corresponding to the normal (top) and attacked (bottom) traffic, which are shown in Fig.4.

Classification was performed for all types of DDoS attacks at once when the following parameters were changed:

- Hurst index for model traffic, that varied from 0.6 to 0.9;
- the level of bursts for model traffic, i.e. the coefficient of variation, that varied in the range from 3 to 14;
- level of attack, that was selected 20, 30 and 40%.

To implement the classification methods, the Python language was used, with libraries that implement machine learning methods. The experiments were carried out using the free Colaboratory platform, which allows you to access powerful computing resources, does not require any customization and is fully functional in the cloud.

Numerical experiments were conducted for model traffic of different lengths and 6 variants of DDoS attacks. Below are the results for time series of 250 values, which is enough for processing data in real time.

As expected, the experiment confirmed that the probability of an attack detecting depends significantly on the attack level. Table 1 shows the average classification accuracy depending on the attack level, obtained for the Hurst parameter H =0.8, the coefficient of variation $\sigma = 7$.

Table 1. Dependence of classification accuracy on attack level

	Attack level		
	20%	**30%**	**40%**
Accuracy	0.93	0.97	0.99

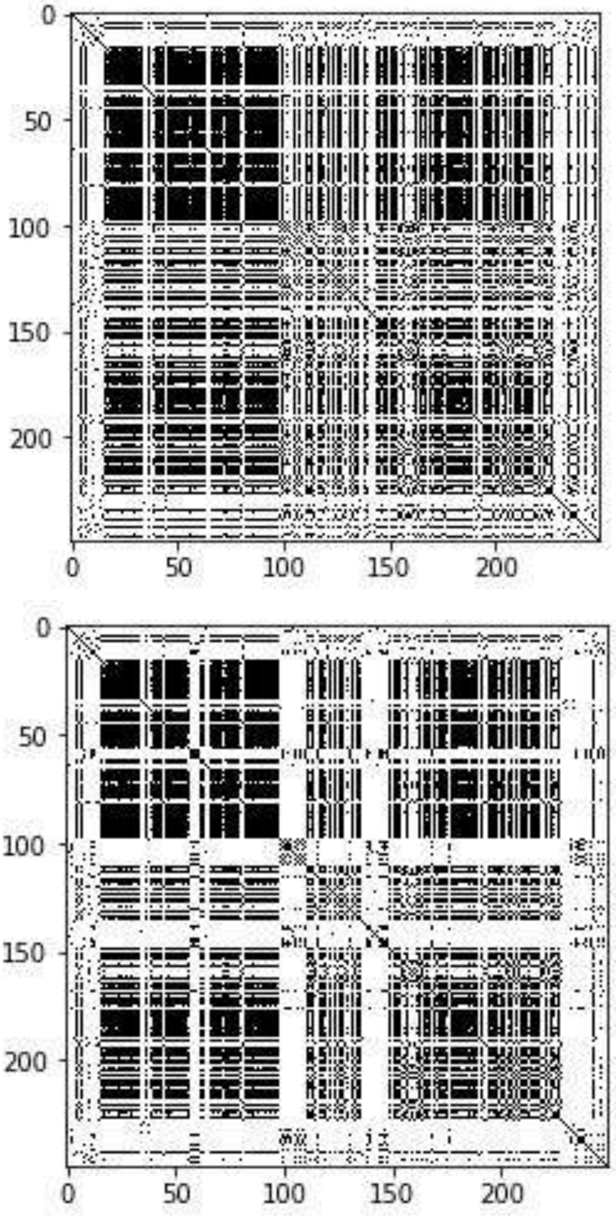

Fig. 6. Recurrence plots for the corresponding traffic realizations presented in Fig. 4.

Table 2 shows the average probability of attack detection if the level of attack is 20%, the coefficient of variation $\sigma = 7$. The values of Hurst index are presented in the range of *H*=0.7-0.9, that corresponds to most of the actual traffic realization. It is obvious that the classification accuracy values decrease with the increase of the traffic Hurst index. This is due to the fact that the attacks realizations have high *H* values, that leads to high *H* values for the total realization [33, 34]. Consequently, the correlation structure of realizations and images of recurrence plots of the attacked and non-attacked traffic will differ greatly in the case of traffic with small values of *H*. The obtained results are in good agreement with the results obtained in [25].

Table 2. Dependence of classification accuracy on traffic Hearst index

	Hurst index		
	0.7	**0.8**	**0.9**
Accuracy	0.97	0.93	0.78

The classification accuracy values depending on the variation coefficient σ are shown in Table 3. The attack level in this case is 20%, and the values of Hurst *H*=0.8. It should be noted that the probability of an attack detection significantly depends on the coefficient of traffic variation and increases with its increase. This can be explained by the fact that high values of the variation coefficient correspond to large bursts, that increases the heterogeneity of recurrence plots and improves image recognition.

Table 3. Dependence of classification accuracy on traffic variation coefficient

	The variation coefficient		
	3	**7**	**13**
Accuracy	65	93	97

Conclusion

The method of detecting DDoS-attacks based on the visualization of recurrence plots and subsequent images classification was proposed in this paper. The attacked traffic realization was obtained by summing up the model of traffic and DDoS-attacks realizations. Residual neural networks were used as a classifier.

The results have shown that the described method has a rather high classification accuracy even at a small level of attack. The analysis showed that the accuracy of the attack detection depends significantly on the self-similarity of the traffic being attacked: the less self-similar the traffic, the easier it is to detect the attack. The probability of intrusion detection also depends on the heterogeneity of the traffic: the higher the amount of bursts, the higher the accuracy of the attack detection.

Our future research will focus on building and training a neural network to detect different types of attacks using real-world traffic data sets.

References

1. Shipmon, D. T., Gurevitch, J. M., Piselli, P. M., Edwards, S.: Time Series Anomaly Detection: Detection of Anomalous Drops with Limited Features and Sparse Examples in Noisy Highly Periodic Data, Google, Inc. Cambridge, MA, USA, pp.1-9 (2016).
2. Zadeh, J.: Time Series Anomaly Detection in Network Traffic: A Use Case for Deep Neural Networks, https://jask.ai/time-series-anomaly-detection-in-network-traffic-a-use-case-for-deep-neural-networks, last accessed 2019/02/11.
3. Kirichenko, L., Radivilova, T.: Analyzes of the distributed system load with multifractal input data flows. 2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM) Proceedings, pp. 260-264. (2017). doi: 10.1109/CADSM.2017.7916130
4. Bernacki, J., Kołaczek, G.: Anomaly Detection in Network Traffic Using Selected Methods of Time Series Analysis. I. J. Computer Network and Information Security 9, 10-18 (2015).
5. Bulakh, V., Kirichenko, L., Radivilova, T. Classification of Multifractal Time Series by Decision Tree Methods. Proceedings of the 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume I: Main Conference 2105 (2018).
6. Radivilova, T., Kirichenko, L., Bulakh, V.: Detection of DDoS-attacks by mashing-lening methods based on fractal properties. Security in cervatury, the social internet space in context values and hazards 1, 299-315 (2019).
7. Detecting Anomalies with Moving Median Decomposition, https://anomaly.io/anomaly-detection-moving-median-decomposition/, last accessed 2019/01/11.
8. Grubbs, F.: Procedures for Detecting Outlying Observations in Samples. Technometrics. Technometrics 11(1), 1–21 (1969). doi:10.2307/1266761
9. Gupta, N., Srivastava, K., Sharma, A.: Reducing False Positive in Intrusion Detection System: A Survey. International Journal of Computer Science and Information Technologies 7(3), 1600-1603 (2016).
10. Esling, P., Agon, C.: Time series data mining. ACM Computing Surveys 46(1) (2012).
11. Ben, D.: Feature-based time-series analysis, https://arxiv.org/abs/1709.08055 last accessed 2019/10/28.
12. Eckmann, J.P., Kamphorst, S.O., Ruelle, D. Recurrence Plots of Dynamical Systems. EPL (Europhysics Letters) 4(9), 973-977 (1987).
13. Marwan, N., Wessel, N., Meyerfeldt, U., Schirdewan A., Kurths, J.: Recurrence-plots-based measures of complexity and application to heart-rate-variability data. Physical Review E 66(2), 026702-1 - 026702-6 (2002).
14. Marwan, N., Romano, M., Thiel, M., Kurths, J.: Recurrence plots for the analysis of complex system. Physics Reports 438(5-6), 237-329 (2007).
15. Kirichenko, L., Radivilova T., Bulakh, V.: Classification of Fractal Time Series Using Recurrence Plots. 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T) Proceedings, pp. 719-724. IEEE (2018). doi: 10.1109/INFOCOMMST.2018.8632010
16. Thilo, M., Spiegel, S., Albayrak, S.: Time Series Classification using Compressed Recurrence Plots http://www.dai-labor.de/fileadmin/Files/Publikationen/Buchdatei/Published.pdf last accessed 2019/10/20
17. Hatami, N., Gavet, Y., Debayle, J.: Bag of Recurrence Patterns Representation for Time-Series Classification, https://arxiv.org/abs/1803.11111v1 last accessed 2019/10/20.

18. Hatami, N., Gavet, Y., Debayle, J.: Classification of Time-Series Images Using Deep Convolutional Neural Networks, https://arxiv.org/abs/1710.00886 last accessed 2019/10/20
19. Takens, F., Rand, D.A., Young, L.-S.: Detecting strange attractors in turbulence. Dynamical Systems and Turbulence: Lecture Notes in Mathematics, Springer-Verlag 898, 366–381 (1981).
20. Kirichenko, L.O., Kobitskaya, Y., Habacheva, A.: Comparative Analysis of the Complexity of Chaotic and Stochastic Time Series. Radioelectronics 2(31), 126-134 (2014).
21. Popa, S.M., Manea, G.M.: Using Traffic Self-Similarity for Network Anomalies Detection. 20-th International Conference on Control Systems and Computer Science Proceedings, pp. 639-644. IEEE (2015). doi: 10.1109/CSCS.2015.89
22. Kaur, G., Saxena, V., Gupta, J.: Detection of TCP targeted high bandwidth attacks using self-similarity. Journal of King Saud University - Computer and Information Sciences (2017). https://doi.org/10.1016/j.jksuci.2017.05.004
23. Deka, R., Bhattacharyya, D.: Self-similarity based DDoS attack detection using Hurst parameter. Security and Communication Networks 9(17), 4468-4481 (2016). doi: https://doi.org/10.1002/sec.1639
24. Kirichenko, L., Radivilova, T., Bulakh, V.: Machine Learning in Classification Time Series with Fractal Properties. Data 4(1) 5, 1-13 (2019). doi:10.3390/data4010005
25. Kirichenko, L., Radivilova T., Bulakh V.: Binary Classification of Fractal Time Series by Machine Learning Methods. In: Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2019. Advances in Intelligent Systems and Computing, vol 1020. Springer, Cham 701-711 (2020). doi: https://doi.org/10.1007/978-3-030-26474-1_49.
26. Bulakh, V., Kirichenko, L., Radivilova, T.: Time Series Classification Based on Fractal Properties. 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) Proceedings, pp. 198-201. (2018). doi: 10.1109/DSMP.2018.8478532
27. Kirichenko, L., Radivilova, T., Alghawli, A. S.: Mathematical simulation of self-similar network traffic with aimed parameters. Annals Computer Science Sériés 11(1), 17-22 (2013).
28. Al-kasassbeh, M., Al-Naymat, G., Al-Hawari, E.: Towards Generating Realistic SNMP-MIB Dataset for Network Anomaly Detection. International Journal of Computer Science and Information Security (IJCSIS) 14(9), 1162-1185 (2016).
29. LeCun, Y., Bengio, Y.: Convolutional Networks for Images, Speech, and Time-Series, in Arbib, M. A. (Eds), The Handbook of Brain Theory and Neural Networks, MIT Press (1995).
30. Dan, C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, High Performance Convolutional Neural Networks for Image Classification. Twenty-Second International Joint Conference on Artificial Intelligence Proceedings, 2, pp.1237–1242. (2013). http://people.idsia.ch/~juergen/ijcai2011.pdf last accessed 2019/10/20
31. Fung, V.: An Overview of ResNet and its Variants https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035 last accessed 2019/10/20
32. Gao, H., Zhuang, L., Weinberger, K. Q., Laurens, M.: Densely Connected Convolutional Networks. arXiv:1608.06993 (2016).
33. Ivanisenko, I., Kirichenko, L., Radivilova, T.: Investigation of self-similar properties of additive data traffic. 2015 X International Scientific and Technical Conference “Computer Sciences and Information Technologies” (CSIT) Proceedings, pp. 169–171. IEEE (2015). doi:10.1109/STC-CSIT.2015.7325459
34. Ivanisenko, I., Kirichenko, L., Radivilova, T.: Investigation of multifractal properties of additive data stream. 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), pp. 305-308. (2016). doi: 10.1109/DSMP.2016.7583564