Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 877–882
DOI: 10.15439/2023F6381
ISSN 2300-5963 ACSIS, Vol. 35

Use of traffic sampling in anomaly detection for high-throughput network links

Marek Bolanowski, Andrzej Paszkiewicz
0000-0003-4645-967X
0000-0001-7573-3856
Rzeszów University of Technology, Rzeszów, Poland
Email: {marekb, andrzejp}@prz.edu.pl

Hubert Mazur
0009-0003-6784-8428
Rzeszów University of Technology, Rzeszów, Poland
Email: hub.mazur99@gmail.com

***Abstract*—Currently, anomaly detection is an increasingly important issue in terms of research work and applications in production systems. Information about system malfunction allows the implementation of precise diagnostic and corrective actions. Two main approaches based on statistical analysis and machine learning techniques are used in anomaly detection systems, which are computationally complex, especially when dealing with high traffic volumes in computer network. In this paper, the limitation of the sampling frequency for network traffic parameters is proposed as a technique to reduce the computational complexity of anomaly detection methods. The proposed approach has been verified in a real network link monitoring system for a medium-sized ISP. The results obtained are promising and can be used to build a production system that enables the development of early warning systems in the area of security incident detection dedicated to high-speed access links.**

I. Introduction

DISTRIBUTED information systems are becoming increasingly prevalent in critical areas of human life. For instance, they are used to control traffic in the city [1], [2], monitor patients' vital signs [3], or manage technological processes in smart factories [4]. This information systems are exposed to a number of new types of cyber security threats. The market offers ready-made tools for executing attacks, which affects the constant increase in the number of security incidents. During the pandemic period alone, cybercrime increased by 600% [5], and the average cost of a data security breach in the U.S. in 2022 was 4.35 million [6]. There is no single effective system of protection against these threats. Nowadays threat detection and elimination systems have a cascade structure. In other words, we have many interconnected layers in which IDS, IPS, ACL, etc. function. Each type of layer is sensitive to different types of attacks. In the case of carrier access links, such as those used for Internet Service Provider (ISP) companies, simple Access Control List (ACL) rules that filter network traffic based on source and destination addresses are generally applicable. Even in the case of such a simple mechanism, the implementation of a larger number of ACLs, or the implementation of a mechanism for logging information (what flow and by what ACL was blocked) can bring significant delays in the transmission path. Therefore, the authors posed the question during their research: is it possible to detect anomalous behavior without introducing additional delay while reducing the computational complexity of detecting process? Anomaly detection is an important data analysis task that detects anomalous or abnormal data from a given data set. Preliminary research has shown that a conducted cyberattack can affect the change of statistical characteristics of network traffic in the access link. Therefore, the analysis of descriptive link parameters, statistical techniques or artificial intelligence can be used in the area of an access link on the border of the protected network to detect the threat. Anomaly detection is widely used in myriad fields such as medical, public health, fraud detection, intrusion detection, industrial damage, image processing, sensor networks, robot behavior and astronomical data [7]. Current research is concerted around speeding up the detection process reducing the computational complexity of the entire process and identifying not only the occurrence of a given anomaly but also eliminating its causes.

At present, there is a clear trend related to identifying the best AI models for anomaly detection in ISP links in order to achieve the best possible detection performance. Applications in this area include both supervised and unsupervised methods [8], [9], [10], [11]. Of course, previously, network traffic sampling methods [12] were used for anomaly detection using traditional IDS probes. Such methods were applied, for example, in the work [9], and the obtained results look promising. Their applications allow for preliminary verification in terms of detecting anomalies in large volumes of network traffic. However, it should be noted that a large body of work in this field is based on previously prepared test datasets [13], [14], [15] or on data obtained from real links with low throughputs [16]. Preliminary results of conducted research have shown that, in addition to data sampling, the proper preparation of acquired data and flow aggregation have a positive impact on detection outcomes. Of course, data preprocessing can also be computationally complex, but it can be easily parallelized and computed distributed among system nodess [17], [18]. The analysis of available literature clearly demonstrates the pursuit of increasing the accuracy of predictive models, but we must not forget about their applicability in real computer networks. In this study, the authors decided to investigate the impact of data set impoverishment (sampling) on the sensitivity of the anomaly detection model and whether it

is possible to limit the number of processed traffic samples while maintaining the detection level. The entire study was conducted in a production network of an ISP (Enf sp. z o.o). The developed detection layer at the ISP access link can serve as an additional layer of protection against cyber-attacks in cascade anomaly detection systems [19]. If the detection effectiveness of the model slightly decreases with decreasing traffic sampling frequency, it will positively contribute to reducing the amount of necessary measurement data to be transmitted and the processing time required, thus increasing the applicability of the solution in real networks.

The article has the following structure: Chapter 2 presents the network structure of the ISP access node and the architecture of the data acquisition and processing system. In Chapter 3, the data aggregation and sampling process are discussed in detail. Chapter 4 describes the model used for anomaly detection. Chapter 5 presents the obtained results, including the accuracy of detection in relation to the sampling frequency. In Chapter 6, the obtained results were summarized, and directions for further research were indicated.

II. ISP EDGE NODE TOPOLOGY

As mentioned earlier, the research was conducted in the environment of a medium-sized ISP. Real network traffic from end customers was analyzed. In order to carry out the research, it was necessary to modify the structure of the access node used in the system. The system structure is shown in Figure 1. The access router (Extreme MLX-4) connect the entire network segment to the Internet using the BGP protocol. The core of the access network was built based on two switches: Extreme 690 (CORE switch) and Extreme 670 (S1 switch). Policy shaping and NAT for the LAN segment were implemented through a software router (TC + IPTables) built on a Dell R710 server. Two additional hosts, PC1 and PC2, were introduced into the network. PC1 was connected to the LAN network using a Dasan switch, while PC2 was connected through a TP-Link switch in the demilitarized zone of the access node. Its task was to emulate an attack on PC1. All traffic transmitted to the LAN is directed through port P1. Using the port mirroring mechanism, the traffic from port P1 is copied to the Dell PowerEdge R940 server, where calculations related to anomaly detection are performed. This server had the following specifications: Intel(R) Xeon(R) Gold 624 CPU @ 2.60GHz processor; 128 GB of RAM; NVIDIA Tesla V100-PCIE-16GB GPU; HDD 4.5 TB. The 'PowerEdge R940' server hosted a virtual machine based on the Debian OS, which collected traffic (bidirectional) using tcpdump. The laboratory setup allowed for data collection in the infrastructure from the layer 1 to layer 7 of the ISO/OSI model, capturing individual packets for specific network flows using the tcpdump sniffer. Such an environment allowed for testing various data processing techniques and AI algorithms to determine the optimal sampling frequency at which the created models would effectively detect abnormal periods in the packet flow in the investigated network.

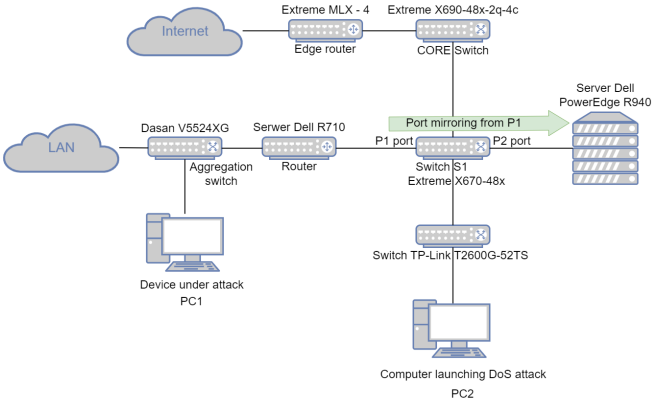

Fig. 1. ISP network edge node architecture with testbed elements

In the next step, a system was built to allow smooth frequency sampling changes. It should be noted that during the conducted research, the entire traffic from port P1 was collected. The entire sampling process was performed on the PowerEdge R940 server, enabling repeated tests for different sampling frequencies. Ultimately, in production systems, the sampling frequency can be set on a specific probe installed in the network. This not only reduces the amount of processed data but also limits the amount of data transmitted between the probe and the detection system. Additionally, initial data preprocessing can also be performed on the measurement probe (in the test system port P1 acts as the probe). Sequential packet selection with a fixed period between consecutive samples was used in the sampling process. In other words, all collected packets were labeled with consecutive natural numbers, and only those packets whose indexes were multiples of a selected natural number s, such as $s = 2$ (sampling every other packet), were chosen for further analysis. Of course, it is possible to apply a different statistical distribution of samples, which will be the subject of further research. The data received from the ISP network was saved in .dump file format. Subsequently, it was divided into equal time intervals (windows). Each window represents a short time of network operation that is evaluated by the machine learning model to classify the entire window as either anomalous or not. The order of the sampling and windowing processes is interchangeable. In the next step, CICFlowMeter software [20] was used for feature extraction. As a result of its operation, CSV files containing feature vectors describing each analyzed packet were obtained. These files were used in further analysis for feature selection and aggregation, which will be described in detail in the subsequent part of the article. The data processing process is described in Figure 2.

In order to describe the process of windowing, i.e., to divide packets into windows depending on the time of their capture, let us make the following assumptions:

$$T = \{t_0, t_1, t_2, \ldots, t_z\}$$
$$t_k = k \cdot f, \quad k \in \mathbb{N}_0, \quad k \leq z$$

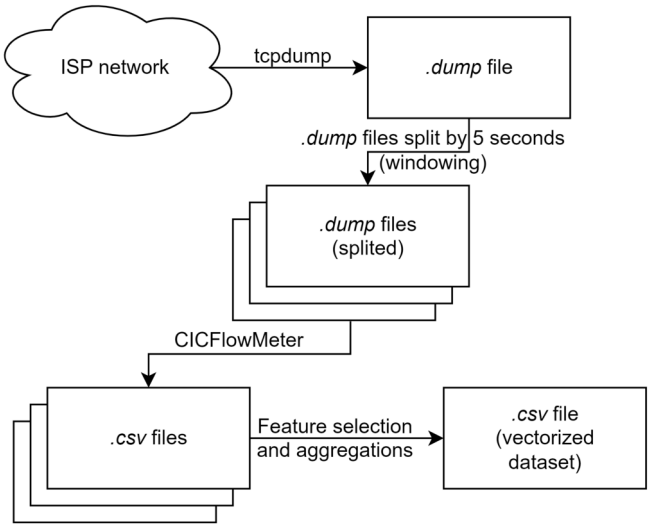

Fig. 2. Data processing scheme

$$z = \left\lfloor \frac{t_{test}}{f} \right\rfloor,$$

where: t_{test} – total duration of the test; k – the number of the given window; P - set of all packages; T – set of all the moments of time in which the windows begin; f – the length of the window within which the packages will be aggregated.

In view of this, we can assume that the set of packages contained in a given window can be described as follows:

$$O_u = \{p \in P : t_u \leq p^{(t)} < t_{u+1}\}$$

$$u = 0, 1, \ldots, z-1$$

where: O – set of all windows; $p^{(t)}$ - packet capture time p.

The data was divided into two sets:

1) The training set represented normal network traffic and was collected for one hour under standard network operating conditions. It consisted of traffic from LAN clients and PC2 (see Figure 1). These data will be used to train a model for the purpose of identifying normal traffic. The anomaly detection model used in the further part of this work will be based on a set of unsupervised algorithms. This approach was chosen because in case of supervised learning model, staff would have to label which packets belonged to normal traffic and which were considered anomalous. This process is extremely time-consuming. Naturally, in the case of unsupervised learning, during the training period, it is essential to ensure that the network is not under attack. Therefore, the training time of the models must be closely monitored by the technical personnel. After training the model on attack-free traffic, it should be able to determine whether incoming packets grouped in windows O_u will contain flows characterized by parameter values deviating from the characteristics of normal traffic. The training dataset contained information on 1,182,566,238 packets.
2) The test set aimed to verify the performance of the model based on the training set. The packets in the windows represented network traffic in two states: normal and anomalous. The anomaly was a 5-minute long Denial-of-Service (DoS) attack. The test dataset contained information on packets captured over a period of 45 minutes, out of which 20 minutes represented normal traffic, the next 5 minutes included the anomaly, and the remainder consisted of normal traffic again. For this dataset, window labeling was performed to mark them as either anomalous or non-anomalous in order to assess the quality of the trained model. The test set contained information on 697,871,782 packets.

It should be noted that during the conducted research, a series of experiments related to DoS and DDoS attacks were carried out, and repeatability of the obtained results was achieved. The DoS attack was identified by the ISP operator as the most common type of attack that the network encounters during its normal operation. Of course, the model shows sensitivity to other types of anomalies not related to DoS attacks, but research in this area needs to be continued.

III. Pre-processing of data

The data collected during the experiments were continuously subjected to the process of cleaning and preparation for further stages of processing related to model training and anomaly detection. According to the scheme presented in Figure 2, all extracted windows O_u had to undergo a vectorization process, so that each window represented independent feature vectors. The vectorization method used in this work is an aggregation approach of selected flow features obtained through feature extraction using the CICFlowMeter software for unique source and destination IP address pairs. A flow represents the packet flow between two network devices, defined by source and destination IP addresses, as well as used ports and network protocols. For the purpose of this work, the notations $p^{(s)}$ and $p^{(r)}$ were adopted to denote the source and destination IP addresses of a given packet, respectively. Therefore, the vectorization process can be described as follows:

$$D^{(u)} = \left\{ F_2\left(F_1\left(R_i^{(u)}\right)\right) : i = 0, 1, \ldots, \left|R^{(u)}\right| \right\},$$

$$R^{(u)} = \left\{ \left\{ p \in O_u : \left\{p^{(s)}, p^{(r)}\right\} = \bar{U}_j^{(u)} \right\}, \right.$$
$$\left. j = 0, 1, \ldots, \left|\bar{U}^{(k)}\right| \right\},$$

$$U^{(k)} = \left\{ \left\{p^{(s)}, p^{(r)}\right\} : p \in O_k \right\},$$

where: $D^{(u)}$ - the aggregated feature vectors of window flows u; $R^{(u)}$ - a set of packet collections with unique destination and recipient IP addresses; $U^{(u)}$ - a set of all destination and source IP address pairs in the window k; $\bar{U}^{(u)}$ - a subset contained in $U^{(u)}$ composed only of its unique elements; F_1 - the first aggregation function, its task is to aggregate packet features for each unique flow; F_2 - the second aggregation

function, its task is to aggregate flow features for each unique destination and recipient IP address pair.

In the first stage (aggregation F_1), the characteristics of each flow occurring in the processed window were aggregated. The set of packets in the window is divided into subsets, where each subset contains the set of packets responsible for the creation of a particular flow. In the second stage, the aggregated characteristics obtained in stage F_1 were further aggregated for each unique destination and recipient IP address pair $p^{(s)}, p^{(r)}$ in the processed window O_u. Additionally one dimension describing the number of flows for unique destination and recipient IP address pairs was added to the final vectors $D^{(u)}$. This type of aggregation allows for a complete vector representation of flow data for a given window, which directly translates into reducing the computational complexity of the detection process by reducing the number of features to 25. These features were selected through experimental work aimed at identifying characteristics that maximize the effectiveness of anomaly detection. The list of all used features is presented in Table I, which also indicates the actions performed in the individual aggregation stages F_1 and F_2.

TABLE I
FEATURES USED IN FEATURE EXTRACTION PROCESS

ID	Feature Description	Aggregation F_1	Aggregation F_2
0	Number of flows		Count
1	Flow duration		Average
2	Number of packets sent	Count	Sum
3	Number of packets received	Count	Sum
4	Total length of packets sent	Sum	Sum
5	Total length of packets received	Sum	Sum
6	Minimum length of packets sent	Minimum	Average
7	Maximum length of packets sent	Maximum	Average
8	Average length of packets sent	Average	Average
9	Standard deviation of length of packets sent	Standard deviation	Average
10	Minimum length of packets received	Minimum	Average
11	Maximum length of packets received	Maximum	Average
12	Average length of packets received	Average	Average
13	Standard deviation of length of packets received	Standard deviation	Average
14	Packets per second	Average	Sum
15	Bytes per second	Average	Sum
16	Packets sent per second	Average	Sum
17	Packets received per second	Average	Sum
18	Minimum packet length	Minimum	Average
19	Maximum packet length	Maximum	Average
20	Average packet length	Average	Average
21	Standard deviation of packet length	Standard deviation	Average
22	Average packet size	Average	Average
23	Average segment size of sent packets	Average	Average
24	Average segment size of received packets	Average	Average

IV. MODEL DESCRIPTION

To test the performance of the sampling frequency's impact on anomaly detection accuracy, a densely connected neural network based on an autoencoder architecture was used [21]. The application of this model for anomaly detection is well-known in the literature, and its effectiveness for the complete dataset was experimentally confirmed in the initial stage of the conducted research. The operation of the adopted model can be divided into two main stages:

1) The forward propagation stage of the neural network, which consists of two key components:
   a) Compression of the input feature vector into fewer dimensions (encoding).
   b) Reconstruction of the compressed feature input vector (decoding).
2) The stage of calculating the reconstruction error based on the comparison of the input vector with the output of the neural network. Based on the reconstruction error, a decision is made to classify the sample into normal or containing an anomaly.

Let M denote the reconstruction error for a single vector w. It can be observed that as a result of applying aggregation F_2, we obtain a set of vectors describing the features of all unique sender and receiver IP address pairs. Therefore, the reconstruction error for a single vector w can be expressed as follows:

$$M_w^{(u)} = \frac{\sum_{i=0}^{24}(D_i^{(u,w)} - m(D^{(u,w)})_i)^2}{25}$$

To calculate the reconstruction errors for all vectors in a given window O_u, the above formula should be applied to each $w = 0, 1, \ldots, \left|D^{(u)}\right|$.

The classification of a window can be expressed as follows:

$$a_u = \begin{cases} \text{anomaly} & \text{if } \max(M^{(u)}) > y \\ \text{no anomalies} & \text{otherwise,} \end{cases}$$

where: y – classification threshold; a_u – window classification decision u; $m(D^{(u,w)})$ - vector reconstructed using autoencoder.

Table II presents the detailed architecture of the utilized autoencoder, which was developed based on conducted experiments aiming to maximize the effectiveness of anomaly detection. The dimensions of the input data to each of the layers is marked as follows: the first dimension marked "-" is the number of feature vectors, which can be arbitrary. The second dimension is the size of the input vectors. The output dimension column describes the dimension of the vectors after calculating the total excitation of each neuron and applying the activation function.

The model was trained using windows from the training dataset. It was trained for 30 epochs using the ADAM[22] optimization method and mean squared error (MSE)[23] as the

TABLE II
AUTOENCODER ARCHITECTURE

Layer	Input dimension	Number of neurons	Output dimension	Activation function.
densely connected	(-, 25)	13	(-, 13)	RELU
densely connected	(-, 13)	6	(-, 6)	RELU
densely connected	(-, 6)	13	(-, 13)	RELU
densely connected	(-, 13)	25	(-, 25)	no activation function

reconstruction loss for window characteristics. Additionally, to improve the weight fitting process, the data underwent standardization[24] using the mean and standard deviation of the features from the windows in the training dataset.

V. RESULTS

The combination of processing data using aggregation of unique sender and receiver IP address pairs, along with a model based on maximum reconstruction error of processed feature vectors in each pair's window, yielded good results in anomaly detection task. The windows where anomalies occurred showed significantly higher maximum reconstruction error compared to those characterized by normal traffic. Table III presents the results of anomaly detection quality on the test dataset. The performance of the developed model was

TABLE III
RESULTS OF MODEL EVALUATION ON THE TEST SET

Sampling frequency (s)	Window size in seconds	Detection accuracy
1	5	100.0%
2	5	100.0%
5	5	100.0%
10	5	100.0%
25	5	99.8%
50	5	87.6%

evaluated on the test dataset for different sampling frequencies $s = 10, 25, 50$. The obtained results are presented in Figures 3 to 5. The maximum reconstruction error for the non-anomalous sender and receiver IP address pair is indicated in blue color, while the reconstruction error for the attacking device's IP address and the target IP address is shown in red color.

The results show that satisfactory performance is achieved even in the case of $s = 25$, which means checking every 25th network traffic sample. It is important to note that in the experiments, the window length was 5 seconds and the entire attack lasted 5 minutes. It is assumed that for longer-lasting attacks with higher network traffic intensity, such as DDoS attacks, the sampling frequency can be further reduced. The sampling threshold should be determined individually based on the characteristics of the specific network and the sensitivity of the system expected by the ISP operator.

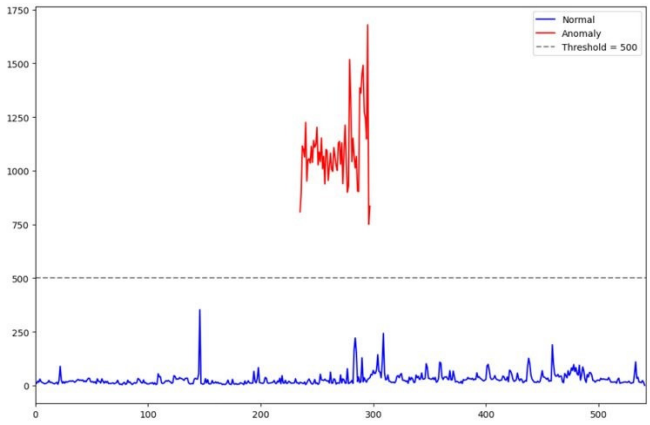

Fig. 3. Model evaluation on test set for sampling every 10 packet

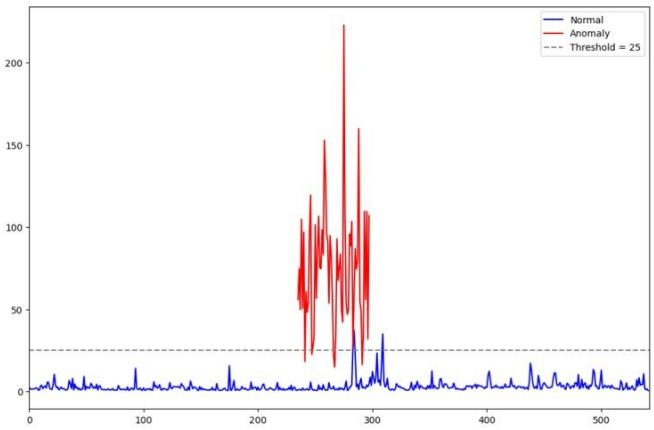

Fig. 4. Model evaluation on test set for sampling every 25 packet

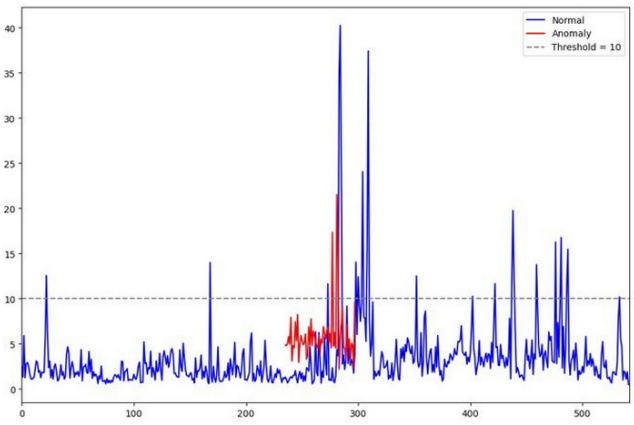

Fig. 5. Model evaluation on test set for sampling every 50 packet

VI. Summary

The paper presents the results of research related to the possibilities of applying a data sampling mechanism for anomaly detection on high bandwidth network links. The research work was carried out in a medium ISP environment in a production infrastructure. The anomaly detection approach proposed in the work taking into account windowing and data sampling allowed to reduce the data needed for anomaly detection (DoS Attack) by 25 times. This makes it possible to reduce the bandwidth of IDS and IPS probes detecting threats, which will directly translate into the cost of implementing cybersecurity systems. Further research concert around the use of non-uniform sequential sampling of traffic, e.g. by using different frequencies and statistical distributions depending on the time of day or network activity. In addition, preliminary studies have shown that the designed system is also effective in detecting other types of anomalies, e.g. data generated by faulty network interfaces. It should be noted that the proposed approach makes it possible to monitor high-throughput access links of ISPs and thus introduce another layer of protection for the entire ICT system against cyber attacks. Thanks to the use of traffic copies, the proposed architecture itself does not bring delays to the end user traffic forwarding process, and once a threat is detected, a given flow can be redirected for further inspection using policy-based routing mechanisms.

Acknowledgment

Work of Marek Bolanowski and Andrzej Paszkiewicz is financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2023. Project number 027/RID/2018/19, amount granted 11 999 900 PLN. The research was carried out in cooperation with "Centrum Badawczo-Rozwojowym Inteligentnych Sieci CBRIS" Enf Sp. z o.o.

References

[1] B. Pawłowicz, M. Salach, and B. Trybus, "Infrastructure of RFID-based smart city traffic control system," in *Automation 2019*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Springer International Publishing, 2020, vol. 920, pp. 186–198. ISBN 978-3-030-13272-9 978-3-030-13273-6 Series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-3-030-13273-6_19

[2] B. Pawłowicz, M. Salach, and B. Trybus, "Smart city traffic monitoring system based on 5g cellular network, RFID and machine learning," in *Engineering Software Systems: Research and Praxis*, P. Kosiuczenko and Z. Zieliński, Eds. Springer International Publishing, 2019, vol. 830, pp. 151–165. ISBN 978-3-319-99616-5 978-3-319-99617-2 Series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-3-319-99617-2_10

[3] S. Dash, S. Biswas, D. Banerjee, and A. U. Rahman, "Edge and Fog Computing in Healthcare – A Review," *Scalable Computing: Practice and Experience*, vol. 20, no. 2, pp. 191–206, 2019. doi: 10.12694/scpe.v20i2.1504. [Online]. Available: https://www.scpe.org/index.php/scpe/article/view/1504

[4] M. Kostolani, J. Murin, and S. Kozak, "An effective industrial control approach," 2019-09-26. doi: 10.15439/2019F187 pp. 911–914. [Online]. Available: https://fedcsis.org/proceedings/2019/drp/187.html

[5] "Cyber security statistics the ultimate list of stats data, and trends for 2023," https://purplesec.us/resources/cyber-security-statistics/, accessed: 2023-05-02.

[6] "Cost of a data breach 2022 a million-dollar race to detect and respond," https://github.com/ahlashkari/CICFlowMeter, accessed: 2023-05-02.

[7] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016. doi: 10.1016/j.jnca.2015.11.016

[8] S. Saha, A. Haque, and G. Sidebottom, "Towards an ensemble regressor model for ISP traffic prediction with anomaly detection and mitigation," in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2022. doi: 10.1109/ISNCC55209.2022.9851774. ISBN 978-1-66548-544-9 pp. 1–6.

[9] M. Shajari, H. Geng, K. Hu, and A. Leon-Garcia, "Tensor-based online network anomaly detection and diagnosis," *IEEE Access*, vol. 10, pp. 85 792–85 817, 2022. doi: 10.1109/ACCESS.2022.3197651

[10] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014. doi: 10.1109/SURV.2013.052213.00046

[11] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447–489, 2019. doi: 10.1007/s11235-018-0475-8

[12] B. Tellenbach, D. Brauckhoff, and M. May, "Impact of traffic mix and packet sampling on anomaly visibility," in *2008 The Third International Conference on Internet Monitoring and Protection*. IEEE, 2008. doi: 10.1109/ICIMP.2008.18. ISBN 978-0-7695-3189-2 pp. 31–36.

[13] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization:," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. SCITEPRESS - Science and Technology Publications, 2018. doi: 10.5220/0006639801080116. ISBN 978-989-758-282-0 pp. 108–116.

[14] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 837601, 2008. doi: 10.1155/2009/837601

[15] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. ACM, 2020. doi: 10.1145/3416013.3426457. ISBN 978-1-4503-8120-8 pp. 37–45.

[16] D. Hulskamp and C. Cappo, "Effectiveness assessment of time series models for anomalies detection in real network traffic," in *2022 41st International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 2022. doi: 10.1109/SCCC57464.2022.10000354. ISBN 978-1-66545-674-6 pp. 1–8.

[17] X. Larriva-Novo, M. Vega-Barbas, V. A. Villagrá, D. Rivera, M. Álvarez Campana, and J. Berrocal, "Efficient distributed preprocessing model for machine learning-based anomaly detection over large-scale cybersecurity datasets," *Applied Sciences*, vol. 10, no. 10, p. 3430, 2020-05-15. doi: 10.3390/app10103430

[18] A. Bhandari, K. Kumar, A. L. Sangal, and S. Behal, "An anomaly based distributed detection system for DDoS attacks in tier-2 ISP networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1387–1406, 2021. doi: 10.1007/s12652-020-02208-3

[19] A. Bădică, C. Bădică, M. Bolanowski, S. Fidanova, M. Ganzha, S. Harizanov, M. Ivanovic, I. Lirkov, M. Paprzycki, A. Paszkiewicz, and K. Tomczyk, "Cascaded anomaly detection with coarse sampling in distributed systems," in *Big-Data-Analytics in Astronomy, Science, and Engineering*, S. Sachdeva, Y. Watanobe, and S. Bhalla, Eds. Springer International Publishing, 2022, vol. 13167, pp. 181–200. ISBN 978-3-030-96599-0 978-3-030-96600-3 Series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-96600-3_13

[20] "Cicflowmeter," https://www.ibm.com/security/data-breach, accessed: 2023-05-02.

[21] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2020. doi: 10.48550/ARXIV.2003.05991 Publisher: arXiv Version Number: 2.

[22] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," vol. 2, no. 1, p. 41, 2019. doi: 10.17977/um018v2i12019p41-46

[23] Y. Liu, "Mean square error of survey estimates," in *Encyclopedia of Quality of Life and Well-Being Research*, F. Maggino, Ed. Springer International Publishing, 2021, pp. 1–3. ISBN 978-3-319-69909-7

[24] M. Gal and D. L. Rubinfeld, "Data standardization," 2018. doi: 10.2139/ssrn.3326377