Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 2259–2268

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Detecting anomalies and attacks in network traffic monitoring with classification methods and XAI–based explainability

Łukasz Wawrowski[a], Marcin Michalak[a,*], Andrzej Białas[a], Rafał Kurianowicz[a], Marek Sikora[b], Mariusz Uchroński[c], Adrian Kajzer[c]

[a]*Research Network Łukasiewicz — Institute of Innovative Technologies EMAG ul. Leopolda 31, 40–189 Katowice, Poland*

[b]*Department of Computer Networks and Systems, Silesian University of Technology, ul. Akademicka 16, 44–100 Gliwice, Poland*

[c]*Wroclaw Centre for Networking and Supercomputing, Wroclaw University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland*

Abstract

Assuring the network traffic safety is a very important issue in a variety of today's industries. Therefore, the development of anomalies and attacks detection methods has been the goal of analyses. In the paper the binary classification–based approach to network traffic safety monitoring is presented. The well known methods were applied to artificially modified network traffic data and their detection capabilities were tested. More detailed interpretation of the nature of detected anomalies is carried out with the help of the XAI approach. For the purpose of experiments a new benchmark network traffic data set was prepared, which is now commonly available.

Keywords: anomaly detection, classification, network traffic security, explainable artificial intelligence ;

1. Introduction

The paper concerns research on the network traffic monitoring to improve the detection of cyber–attacks. The majority of cyber–attacks are detected by signature–based methods. These attacks and protection methods against them are commonly known. The related risk can be mitigated by applying a proper security measure. If an attacker, while exploring a vulnerability, can immediately invent a new attack method based on this vulnerability, we face the so called zero-day attack which increases the security problem. It is possible to detect such attacks by analyzing network traffic — particularly looking for anomalies in this traffic. Some anomalies may be caused by atypical network users'

* Corresponding author. Tel.: +48 32 2007 712;
E-mail address: Marcin.Michalak@emag.lukasiewicz.gov.pl

1877-0509
10.1016/j.procs.2021.08.239

operations, such as software update or big files transfer. Others may be caused by known or unknown attacks. To detect anomalies and interpret their meaning, one can use machine learning methods. The paper deals with these methods and is closely related to the anomaly detection module developed in the RegSOC project. RegSOC is a specialized Security Operations Centre (SOC) designed mainly for public institutions. Each SOC [24] is based on three pillars: people, processes and technology. The anomaly detection module, the key technological component of the RegSOC technology, is implemented in NIDS (network–based intrusion detection system).

In general, two approaches for anomaly detection may be taken into consideration. In the common interpretation, an anomaly means something different from something typical. Thus detection of anomalies relies on the observation of cases that differ somehow (and significantly) from the normality. However, the formal model of anomaly is not provided in any terms. But on the other hand, when the patterns of anomalies are known and when the data may be labelled as anomalies and typicalities then the issue of anomaly detection may be performed as a typical classification task where anomalies are just one (or more) case–defined classes. The influence of artificial outlier introduction on the classification model quality was already the object of a previous study [15, 14, 13].

In this paper we try to focus on the second of the above mentioned attempts to anomaly detection. In our experiments we introduced artificially some disturbances to the normal network traffic and later we tried to detect them as we knew in which time periods such anomalies were present. To decrease the class imbalance, we decided to treat each kind of disturbance as the same type of anomaly. However, we expected also to be able to recognize the real nature of a disturbance with the application of the so called Explainable Artificial Intelligence approach.

The paper is organized as follows: it starts with a short overview of the typical anomaly detection techniques and a review of most popular benchmarks referring to network traffic data that led us to prepare the new data set. Next the experimental section describes the testing environment, the modification of the Cross–Validation model for the time series analysis, the description of the new benchmark data and finally provides the results of the experiments. The paper ends with some conclusions, the perspective of further works and the new benchmark localization information.

2. Related works

2.1. Anomaly detection

Despite the common feeling of the meaning of the anomaly notion it is hard to provide a proper definition that would satisfy many aspects of its application. One of the oldest definitions [9] claims that “an outlying observation is one that appears to deviate markedly from other members of the sample in which it occurs”. Several years later [10] a different approach was presented — an outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism. In the work [1] an outlier is defined as the observation that is inconsistent with the rest of data. Such an approach was later [36] explained in a more detailed manner: an outlier is the observation that does not follow the same model as the rest of the data.

All developed anomaly detection methods may be generally divided into two groups: statistical and density based. The first group most often analyzes only one dimension. Due to that, such an approach to multidimensional data requires further post–processing of the obtained results. The raw result for all dimensions requires a logical mapping: whether at least one variable value is pointed to be an outlying value, an assumed percentage of variables behave in such a way or values of all variables are pointed as outlying observations. The most popular methods of this branch are 3σ test, Grubb’s test [1] or the GESD approach [26].

The next group of methods bases on the local data distribution: typical observations should rather gather in such a way that they generate more dense regions of the data space, while outliers should be found without such a popular neighborhood. Many approaches apply the variants of the k–nearest neighbors analysis [25] others generate the local density based ranking of possible outliers [5, 8].

The two mentioned above groups do not cover all developed anomaly detection approaches. One may find a completely different solution. One of them is the modification of a well known classification method, which is Support Vector Machine [4], called One Class SVM [27]. The other state–of–the–art method is the clustering method called DBSCAN [7], which may be also used for anomaly detection. Originally, it tries to find the partition of the set of objects. However, it allows some objects not to be assigned to any group. Such observation may be considered as a

kind of noise/chaos. Such observations may be also interpreted as outliers. Also other approaches to outlier detection may be invoked: [17, 6].

Nowadays, many other approaches to outlier detection may be found in the literature such an applying a local dynamic neighborhood [35], ensamble learning [11], linear regression [22] or finally a deep Taylor decomposition [16].

2.2. Benchmark datasets for anomaly detection

Anomaly detection techniques proposed by researches are usually evaluated using benchmark datasets such as UGR16 [19], UNSW-NB15 [23], KDDCUP99 [34], or NSLKDD [33]. Usually datasets are available in the pcap (tcpdump),csv or flow (netflow) format. The UGR16 dataset was built based on real internet traffic and common attacks. Data were collected from netflow v9 collectors from Spanish ISPs. This dataset contains labeled attacks such as DoS, scans (UDP, SSH), SPAM, and botnet. Records with normal internet traffic are also included. The UNSW-NB15 dataset was proposed in response to the lack of modern datasets representing internet traffic behavior during cybersecurity attacks. This dataset represents contemporary characteristics of internet traffic in the sense of normal and synthesized cybersecurity attacks of the network traffic. Dataset records contain attacks such as Dos, backdoors, fuzzers, analysis, exploit, shellcode and worms. The datasets KDDCUP99 and NSLKDD were generated over a decade ago and generally were an improved version of KDD99. These datasets contain normal traffic and four categories of attacks: DoS, remote to local (where an attacker has no access to the victim machine), user to root (where an attacker has access to the victim machine), probe (attacker tries to obtain information about victim host).

Table 1 presents the comparison of several well known network traffic datasets. They are described with several attributes: "tagging way" means the way of labeling the objects as normal or as abnormal which may be done before (the planned introduction) or after (by an outer system) data capture; # records provides information about the number of objects in the dataset; anomalies fraction presents the ratio of anomalies in the data; attack nature points whether anomalies are artificially introduced or not; real traffic background means whether the whole data set comes just from closed artificial traffic or the artificial traffic is mixed with the background of real traffic, and finally "freshness" refers to the "age" of the dataset — some old ones may not reflect the present characteristic of the network traffic. The comparison of benchmarks described above may be found in Table 1.

Table 1. Comparison of popular network traffic datasets and the new presented one (* — the number of minutely aggregated records on the basis of 187,665,600 flows of duration not longer than 60 seconds.).

dataset	tagging way	number of records	anomalies fraction	attacks nature	real traffic background	"freshness"
NSLKDD	before	173,709	~47 %	artificial	no	old
KDDCUP99	before	5,209,458	~80 %	artificial	no	old
UNSW–NB15	before	2,218,761	14.48 %	artificial	yes	rather new
UGR16	after	~1,690,000,000	~0.06 %	natural	yes	rather new
RegSOC–KES2021	before	12,960*	4.74 %	artificial	yes	new

We gave up on the first two sets because of their age and a very high fraction of anomalies, which does not suit the real network traffic characteristic. Additionally, these sets contain only artificial network description not mixed with the real one. The third set is a bit younger and artificially introduced anomalies are mixed with the real data, however, in our opinion the anomaly frequency was still too high did not correspond to typical situations. All these three datasets have a common and desirable feature — the location of anomalies was known before the data analysis. The fourth set anomalies were tagged after the data acquisition by some external tools. That may introduce some inconsistency in the data in such a way that non-detected anomalies are treated as normal observations. Finally, the newly introduced dataset — RegSOC–KES2021 — combines the best mentioned features: the set comes from the 2021 network traffic monitoring, anomalies are introduced manually so their location is known before the data acquisition, the artificially introduced traffic is mixed with the real one and the fraction of anomalies is on a satisfactory level.

3. Infrastructure, Environment, and Experiments

The experiments, whose results are presented and discussed below, were performed in the inner network infrastructure of a medium–sized enterprise, which is also active in the IT market. The scenario of modeled attacks was prepared by the members of Wroclaw Centre for Networking and Supercomputing. The collected data are publicly available — the details are presented at the end of the paper.

3.1. Environment description

For the purpose of the work a research environment was prepared within the medium–sized enterprise network. The logical scheme of the network components is presented in Fig. 1.

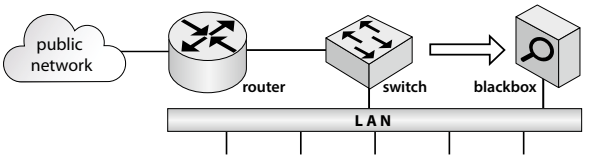

Fig. 1. Diagram with a section of the inner LAN infrastructure, including the elements of the research environment. Blackbox is equipped with two interfaces: LAN, to access the blackbox content and mirroring (arrow) to sniff the incoming/outcoming transfer.

To acquire the network traffic, port mirroring was used on the first switch placed behind the edge router. Such a solution makes it possible to track the whole incoming and outcoming traffic. This seems to be the most important aspect when it is necessary to detect anomalies and potential threats in the traffic. On the other hand, however, the adopted solution makes it impossible to track the whole of the internal traffic in the LAN network — what we can see is only a part of the traffic coming through the monitored switch. In addition, the solution does not allow to observe attacks launched from the outside (we monitor the situation behind the edge router which takes on the external traffic and is the first line of defence). This location of the network traffic measurement was practically the only possible solution to be implemented in the analyzed infrastructure. Moreover, it seems to be an optimal choice as it allows to observe the most important anomaly-detection point and, at the same time, significantly limits the volume of data to be analyzed (we do not register the router–filtered traffic and the majority of the LAN traffic).

The experiments were carried out in the infrastructure presented in Fig. 1. However, from the data flow point of view the more detailed description of components used will be discussed now. The network traffic description data (prepared by the Packetbeat software) were stored in the Elasticsearch environment. This environment participated also in data preprocessing: the original data were limited to flows not longer than 60 seconds and later aggregated into one minute aggregates. The flow was taken as the minute aggregate input if its end time occurred this minute.

Each aggregate consisted of 45 independent variables and a class (normal or anomaly). The set of independent variables was built on the basis of simple network flows characteristics such a number of flows, number of incoming, outcoming packets (and their total), total number of sent, received bytes (and their total) as well as on the basis of the number of specified type of services (dns, ftp, http, icmp, pgsql, rdp, ssh, tls). From the number of packets and number of bytes (in both directions separately as well as analyzed together) six summary statistics were derived: minimum, mean average, quartiles and maximum.

The further analysis was performed in the R environment where four classification methods were used: random forests, neural network, logistic regression and gradient boosting. These models were fitted in the h2o package [18].

3.2. Time series Cross–Validation

It is typical to apply the cross–validation [30] paradigm in a classification task. However, in the case of time series such an approach is not proper as it may occur that later observations participate in building a model applied for earlier ones [2]. Instead, the time series cross–validation (TSCV) was applied (Fig. 2).

In such an approach a single iteration consists on building a model on the train set and evaluate it on the test one, and all train samples are earlier than test samples. The following iteration moves the train objects from the last

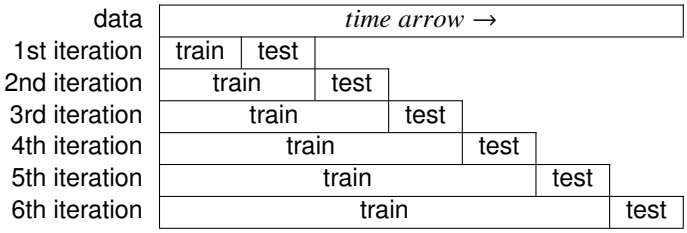

Fig. 2. 6 fold time series cross–validation for time series.

iteration to the current loop run train data and replaces the test data with the newer observations. There is no time conflict in any iteration (tested objects are newer than training) and several models built in the same way are applied on different data (as it is required in the typical cross–validation model).

3.3. Normal network traffic disturbances

It is difficult to provide a proper procedure of anomaly detection software testing. The most crucial issue is to prepare the known — in terms of time and meaning — artificial attacks in the monitored infrastructure. Here two machines imitating attackers located outside the monitored network were used as well as two other machines located inside the network, that were the objectives of attacks. For testing purposes one windows system, that was set up to have three services ftp, rdp and ssh and one linux machine with three configured services: ssh, ftp and http were attacked by the other two linux machines. The attacks were spread over 72h. On both attacked machines inside the infrastructure the normal traffic was created with the JMeter. Normal traffic consisted of browsing Google, Facebook, YouTube as well as sending emails. The first attacks that were used were brute–force attacks on ssh and rdp services on both machines inside the infrastructure. Those were volumetric attacks whose purpose was to generate as much infected traffic as possible. Later, a DOS attack on the linux machine was performed and tests were concluded with brute–force attacks on rdp and ftp services. All attacks had basic purpose to generate visible infected traffic that can be later traced and cross checked with created anomalies. The moments of anomalies introduction and their duration are presented in Fig. 3.

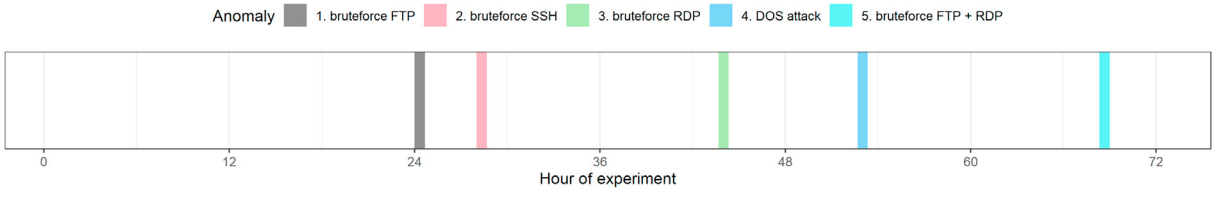

Fig. 3. Moments of anomalies introduction and their duration.

The scenario of attacks (anomalies introduction) was performed three times on different days. Though the same scenario was used, it was mixed with the different background of normal traffic in the network. Thus, the obtained network traffic descriptions were finally different but comparable.

To increase the number of folds in CV, we decided to split the data from the second and third 72 hour network monitoring scenario into four 36 hour smaller sets, used as test sets in the following folds. That finally means, that according to TSCV (Fig. 2) our experiments were four fold. It is worth to be stressed that even after such a split there were two or three introductions in each fold. Four classification methods were used in the experiments: gradient boosting (GBM), logistic regression (GLM), random forests (RF), and neural networks (NN).

3.4. Results and discussion

In this section the results of the network traffic data analysis are presented, starting from the application of different methods on time series data classification. Based on these results, two other methods are taken into further considera-

tion and the most important variables correlation analysis is performed. Finally, the application of XAI for one of the models is carried out.

E**X**plainable **A**rtificial **I**ntelligence (XAI) is an approach started in 1970's [29], also developed later in [32] and [12], and consists on presenting the way why (on the basis of which premises) some AI methods predict the value of the variable of interest. As it was mentioned in our approach to network traffic monitoring, we decided to join all artificially introduced anomalies to one positive class and build a binary classifier for the anomalies detection. After promising results of their detecting we wanted to check whether it is possible — in the case of a suspicious object detection — to provide some more detailed information about the nature of such occurrence. To achieve this aim, Shapley values for predictive models [31] were calculated. The idea of this method comes from the game theory [28] and it shows how model prediction can be distributed across particular variables.

3.4.1. 4–fold Time series Cross–Validation

The split of the data described above allowed to perform a 4–fold time series cross–validation scheme. The averaged confusion matrices for any of four classifiers are presented in Table 2 (for the train data) and in Table 3 (for the test data). As the classes were definitely imbalanced, the balanced accuracy (denoted as "bal. acc." in tables) was used as the classification quality measure. As in each iteration the train set is bigger, it is intuitive to expect that the last model is much wiser than the first one so their accuracies should not be treated in the same way. Each fold model has its weight, which is the ratio of the train set size and the whole data set (from three 72h experiments) size. That led to the following model weights: 0.33 (first model), 0.50, 0.66, and 0.83 (fourth model). The value of the weighted balanced accuracy is denoted in tables as "w. bal. acc.".

Table 2. Averaged results on train sets.

GBM			
prediction	class		
		0	1
	0	2884.2	1.4
	1	0.4	138.0
bal. acc.			0.9953
w. bal. acc.			0.9949

GLM			
prediction	class		
		0	1
	0	2882.8	3.0
	1	1.8	136.4
bal. acc.			0.9884
w. bal. acc.			0.9889

RF			
prediction	class		
		0	1
	0	2884.6	0.3
	1	0.0	139.1
bal. acc.			0.9988
w. bal. acc.			0.9990

NN			
prediction	class		
		0	1
	0	2882.8	2.4
	1	1.8	137.0
bal. acc.			0.9911
w. bal. acc.			0.9911

Despite the high unbalance of the data, we observe quite good classification results on the train data. The balanced classification accuracy does not fall below the level of 98.8%. Classification qualities become not so comparable when test set results are taken into consideration — Table 3.

Table 3. Averaged results on test sets.

GBM			
prediction	class		
		0	1
	0	821.4	0.8
	1	1.6	40.2
bal. acc.			0.9889
w. bal. acc.			0.9900

GLM			
prediction	class		
		0	1
	0	819.9	0.6
	1	3.1	40.4
bal. acc.			0.9905
w. bal. acc.			0.9921

RF			
prediction	class		
		0	1
	0	822.9	8.7
	1	0.1	32.3
bal. acc.			0.8892
w. bal. acc.			0.8963

NN			
prediction	class		
		0	1
	0	822.6	9.4
	1	0.4	31.6
bal. acc.			0.8585
w. bal. acc.			0.8865

It occurs that two methods provide comparable results on test sets and on train data: a small decrease of quality for GBM and even smaller accuracy increase for GLM. In the case of other two methods the level of quality decrease exceeds more than 10 percentage points. Thus, for the further analysis only GBM and GLM models were taken into consideration.

3.4.2. Feature importance and correlation — the post–hoc analysis

For the two best methods (GBM and GLM) it is possible to provide a simple analysis of independent variables importance and their correlation. It was our intention not to analyze variable correlations as the input set consisted of over 40 signals. Here the correlation analysis of the most important variables is presented.

Table 4. The feature importance ranking for the GBM (left) and GLM (right) models.

GBM		
rank	variable	percentage importance
1	ftp	37.67
2	rdp	21.32
3	flow	20.16
4	ssh	20.06
5	network bytes Q3	0.09
6	http	0.08
7	network bytes Q1	0.07
8	dns	0.07
9	destination packets Q3	0.06
10	destination bytes Q3	0.06

GLM		
rank	variable	percentage importance
1	ftp	20.12
2	rdp	18.26
3	flow	17.32
4	ssh	17.11
5	network packets Q3	5.68
6	destination packets Q1	4.76
7	http	2.69
8	destination bytes Q3	2.43
9	network bytes Q2	2.02
10	destination packets Q3	2.01

The variables of the top 10 importance for both models separately are presented in Table 4.

Four variables that deal with a specified port/service were ranked on the highest positions and with the same order. We may notice that ports and services are strongly connected with the nature of attacks planned in the scenarios. Additionally, the disproportion between these four and the remaining variables importance confirms the importance of these variables usage in predictive models. To ensure that the most significant variables are not correlated to each other, in Fig. 4 correlation matrices of variables from Table 4 are presented. Only statistically significant values ($p < 0.05$) are provided — insignificant ones are crossed out.

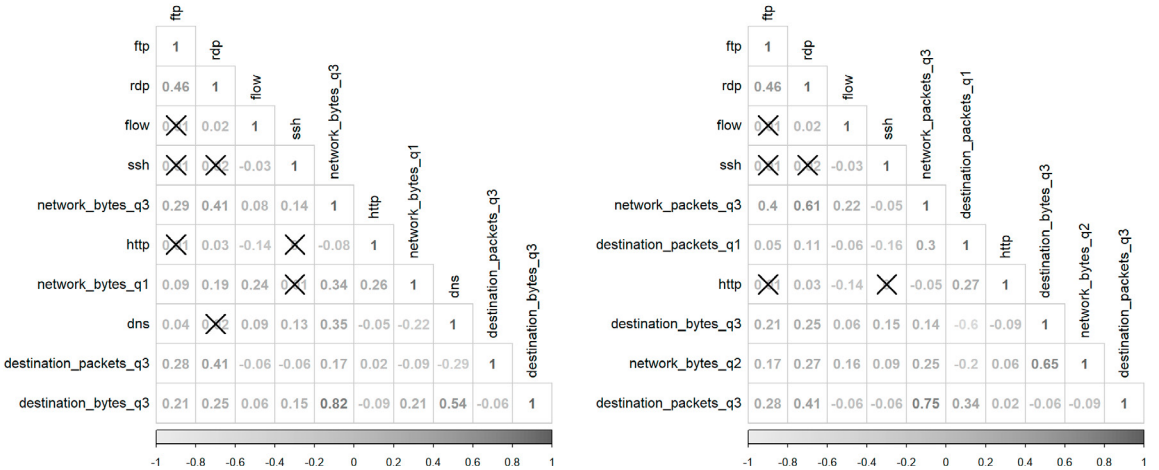

Fig. 4. Linear correlation coefficients for pairs of variables from the GBM model top ten important variables (left) and from the GLM top ten important variables (right).

The absolute values of linear correlation coefficients are generally not so high. For top 4 variables only one pair is characterized by high correlation: ftp and rdp. However, such a situation is easy to explain because two attacks were performed at the same time (fifth anomaly in Fig. 3). Other correlations — even if they are strongly correlated — do not influence the model correctness: high correlation of some variables (e.g. 3rd quartile of network and destination bytes or packets) comes directly from the nature of the data, as network bytes/packets are the total of the source and destination bytes/packets.

3.4.3. Explainability analysis application

We decided to apply the h2o package as it is one of the possible for the R environment. Unfortunately, this package does not support Shapley calculations for linear models. As a result of that only one of two models (gradient boosting) results could be post–processed later with XAI.

To check whether XAI can provide more detailed information about the nature of found anomaly, we selected one observation from each type of attack classified as an anomaly. Then we calculated the SHAP coefficient for each variable which finally participated in the reasoning process. The scale of SHAP contributions for the top 10 variables (as presented in Table 4 — left) for different types of attacks are presented in Figures 5 to 7. A variable name and its value for the specific object are presented on the Y axis labels and they are sorted in a descending order according to the SHAP value.

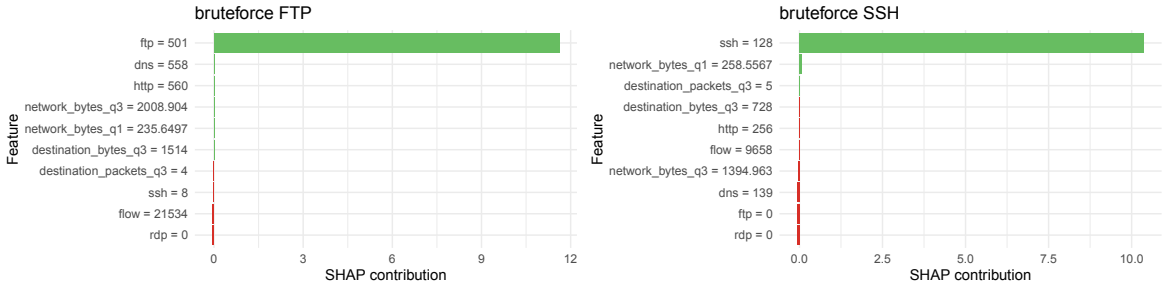

Fig. 5. Results of explainability analysis for the FTP brute–force attack (left) and for the SSH brute–force attack (right).

In the case of the SHAP statistics, for the first two attacks (Fig. 5) we observe high significance of one variable: the number of flows sent through the FTP protocol in the case of FTP brute–force attack and the number of flows sent through the SSH protocol.

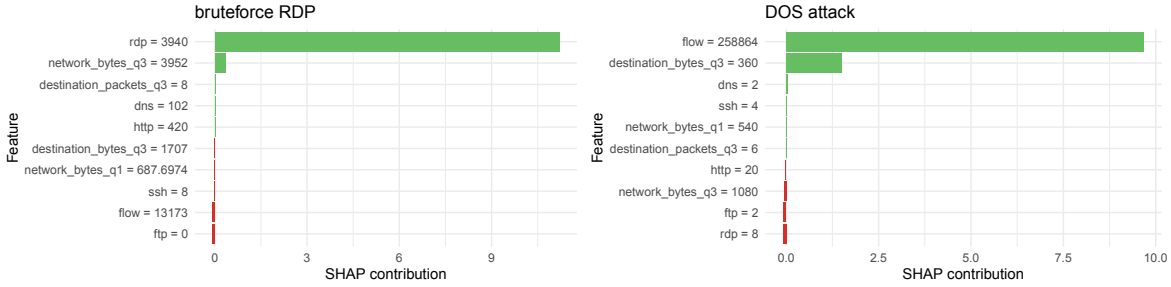

Fig. 6. Results of explainability analysis for the RDP brute–force attack (left) and for the DOS attack (right).

In the case of other two types of attacks (Fig. 6) it is also possible to point the dominating variables (events addressed to the RDP service and the number of flows accordingly). However, it is worth to notice that in the case of the RDP attack the third quartile of the number of network bytes and in the case of the DOS attack third quartile of destination bytes, also distinguish from the rest of variables. This may be a suggestion to analyze such a behavior in future, just for better understanding of the attack behavior.

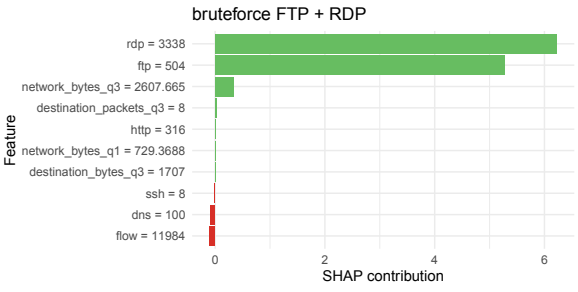

Fig. 7. Results of explainability analysis for the combined FTP and RDP brute–force attack.

Finally, Fig. 7 points that in the case of attacks performed at the same time, two dominating variables are the same as those dominating in separate attacks. That is a very important and significant remark. It means that even performing more than one type of attacks simultaneously it is possible to detect the same network traffic characteristics.

The figures above base on the common variable importance ranking. It is easy to provide the same kind of charts for each attack based just on the object variable SHAP ranking. However, these charts do not provide any new information about the feature influence. For this reason, these figures are not presented.

4. Conclusions and further Works

The paper presents a new approach to anomaly detection. It is dedicated to be applied as a complementary tool for RegSOCs [3]. In opposition to typical anomaly detection methods [21], it bases on binary classification methods application with the further XAI analysis to distinguish between specific anomaly types. This hybrid method is developed to run in already prepared open–source–based architecture [20].

Multiple types of attacks were treated as the common class and four binary classifiers were tested. After cross–validation tests only two of four methods were chosen for further analysis: logistic regression and gradient boosting. To assure that such built models do not rely too much on linearly dependent variables, the correlation analysis of the top 10 important variables was also carried out. Finally, the XAI analysis of the nature of found anomalies confirmed that it is possible to get some more detailed information about the inner type of anomaly.

The data used in the experiments are publicly available. They may be downloaded from the project site http://ibemag.pl/pl/szczegoly-projektow#RegSOC — the Available Datasets section. In case of any problems please contact the corresponding author.

Our feature works will focus on several aspects of the presented data analysis scheme. The first natural goal is to extend the attack scenario with a variety of other detectable anomalies. With such an extension it may become interesting to compare the multi–class models with the presented ones (binary classifier and XAI analysis). As a scenario may consists of several concurrent attacks, it may be also interesting to know whether multi–label classifiers will be appropriate tools. Apart from that, as it was also mentioned, we will try to apply the XAI paradigm also for the second good predictive model — the logistic regression.

Acknowledgements

RegSOC — Regional Center for Cybersecurity (http://regsoc.pl). The project is financed by the Polish National Center for Research and Development as part of the second CyberSecIdent — Cybersecurity and e–Identity competition (agreement number: CYBERSECIDENT/381690/II/NCBR/2018).

References

[1] Barnett, V., Lewis, T., 1994. Outliers in Statistical Data. 3rd ed., Wiley.
[2] Bergmeir, C., Benítez, J.M., 2012. On the use of cross-validation for time series predictor evaluation. Information Sciences 191, 192–213. Data Mining for Software Trustworthiness.
[3] Bialas, A., Michalak, M., Flisiuk, B., 2020. Anomaly detection in network traffic security assurance, in: Zamojski, W., et al. (Eds.), Engineering in Dependability of Computer Systems and Networks, Springer International Publishing, Cham. pp. 46–56.
[4] Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers, in: Haussler, D. (Ed.), Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92), ACM Press. pp. 144–152.
[5] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J., 2000. LOF: Identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, p. 93–104.
[6] Byers, S., Raftery, A.E., 1998. Nearest-neighbor clutter removal for estimating features in spatial point processes. Journal of the American Statistical Association 93, 577–584.
[7] Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press. p. 226–231.
[8] Gao, J., Hu, W., Zhang, Z.M., Zhang, X., Wu, O., 2011. RKOF: Robust kernel–based local outlier detection, in: Advances in Knowledge Discovery and Data Mining, pp. 270–283.
[9] Grubbs, F.E., 1969. Procedures for detecting outlying observations in samples. Technometrics 11, 1–21.
[10] Hawkins, D.M., 1980. Identification of Outliers. Monographs on Applied Probability and Statistics, Springer Netherlands.

[11] Iftikhar, N., Baattrup-Andersen, T., Nordbjerg, F.E., Jeppesen, K., 2020. Outlier detection in sensor data using ensemble learning. Procedia Computer Science 176, 1160–1169.
[12] Johnson, W., 1994. Agents that learn to explain themselves, in: AAAI-94 Proceedings, pp. 1257–1263.
[13] Kalisch, M., Michalak, M., Przystalka, P., Sikora, M., Wróbel, L., 2016a. Outlier detection and elimination in stream data - an experimental approach. Lecture Notes in Computer Science 9920, 416–426.
[14] Kalisch, M., Michalak, M., Sikora, M., Wróbel, L., Przystalka, P., 2016b. Data intensive vs sliding window outlier detection in the stream data - an experimental approach. Lecture Notes in Computer Science 9693, 73–87.
[15] Kalisch, M., Michalak, M., Sikora, M., Wróbel, L., Przystalka, P., 2016c. Influence of outliers introduction on predictive models quality. Communications in Computer and Information Science 613, 79–93.
[16] Kauffmann, J., Müller, K.R., Montavon, G., 2020. Towards explaining anomalies: A deep taylor decomposition of one-class models. Pattern Recognition 101, 107198.
[17] Knorr, E.M., Ng, R.T., 1998. Algorithms for mining distance-based outliers in large datasets, in: Proceedings of the 24rd International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. p. 392–403.
[18] LeDell, E., Gill, N., Aiello, S., Fu, A., Candel, A., Click, C., Kraljevic, T., Nykodym, T., Aboyoun, P., Kurka, M., Malohlava, M., 2020. h2o: R Interface for the 'H2O' Scalable Machine Learning Platform. URL: https://github.com/h2oai/h2o-3. r package version 3.32.0.3.
[19] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., Therón, R., 2018. Ugr'16: A new dataset for the evaluation of cyclostationarity-based network idss. Computers & Security 73, 411–424.
[20] Michalak, M., Wawrowski, L., Sikora, M., Kurianowicz, R., Kozłowski, A., Białas, A., 2022. Open–source–based environment for network traffic anomaly detection, in: Engineering in Dependability of Computer Systems and Networks, p. in press.
[21] Michalak, M., et al., 2021. Outlier detection in network traffic monitoring. 10th Int. Conf. on Patt. Recogn. Appl. and Methods 1, 523–530.
[22] Mondal, M.A., Rehena, Z., 2020. Road traffic outlier detection technique based on linear regression. Procedia Computer Science 171, 2547–2555. Third International Conference on Computing and Network Communications (CoCoNet'19).
[23] Moustafa, N., Slay, J., 2015. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6.
[24] Muniz, J., McIntyre, G., AlFardan, N., 2015. Security Operations Center: Building, Operating, and Maintaining Your SOC. Cisco Press.
[25] Ramaswamy, S., Rastogi, R., Shim, K., 2000. Efficient algorithms for mining outliers from large data sets. SIGMOD Rec. 29, 427–438.
[26] Rosner, B., 1983. Percentage points for a generalized esd many-outlier procedure. Technometrics 25, 165–172.
[27] Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Platt, J., 1999. Support vector method for novelty detection, in: Proceedings of the 12th International Conference on Neural Information Processing Systems, MIT Press, Cambridge, MA, USA. p. 582–588.
[28] Shapley, L.S., 1953. A value for n-person games. Contributions to the Theory of Games 2, 307–317.
[29] Shortliffe, E., 1976. Computer–Based Medical Consultations: MYCIN. Elsevier.
[30] Stone, M., 1974. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society. Series B (Methodological) 36, 111–147.
[31] Strumbelj, E., Kononenko, I., 2010. An efficient explanation of individual classifications using game theory. J. Mach. Learn. Res. 11, 1–18.
[32] Swartout, W., Paris, C., Moore, J., 1991. Explanations in knowledge systems: design for explainable expert systems. IEEE Expert 6, 58–64.
[33] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. doi:10.1109/CISDA.2009.5356528.
[34] University of California, . KDD Cup 99 Dataset. https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed: 2021-04-14.
[35] Wang, R., Zhu, Q., Luo, J., Zhu, F., 2020 (in press). Local dynamic neighborhood based outlier detection approach and its framework for large-scale datasets. Egyptian Informatics Journal .
[36] Weisberg, S., 2005. Applied Linear Regression. Wiley Series in Probability and Statistics. 3rd ed., Wiley & Sons.