Agnieszka DURAJ[1], Paweł BUCKI[1], Aleksander DRAJLING[1], Robert MAKROCKI[1], Mateusz SIPIŃSKI[1]

Institute of Information Technology, Lodz University of Technology (1)

doi:10.15199/48.2022.12.46

Anomaly detection in network traffic

Abstract. *The authors of this paper faced the problem of detecting anomalies, understood as potential attacks in network traffic occurring on a document-signing computing cluster. In an infrastructure exposed to the public world, it is extremely difficult to distinguish traffic generated by users from traffic generated by a network attack. The solution the authors present, based on the collected data, determines whether the traffic from the selected sample originated from an attack or not, based on ready-made clustering algorithms. The performance of the following algorithms was compared: DBSCAN, LOF, COF, ECOD and PCA.*

Streszczenie. *Autorzy niniejszej pracy stanęli przed problemem wykrywania anomalii, rozumianych jako potencjalne ataki, w ruchu sieciowym zachodzącym na klastrze obliczeniowym podpisującym dokumenty. W infrastrukturze wystawionej na świat publiczny niezwykle trudno odróżnić ruch generowany przez użytkowników od ruchu generowanego w ramach ataku sieciowego. Rozwiązanie jakie autorzy przedstawiają na podstawie zbieranych danych określa, czy ruch z wybranej próbki powstał w wyniku ataku czy nie, na podstawie gotowych algorytmów grupowania. Porównano działanie następujących algorytmów: DBSCAN, LOF, COF, ECOD oraz PCA.* ***(Wykrywanie anomalii w ruchu sieciowym)***

Keywords: anomaly detection, network anomaly, attack detection
Słowa kluczowe: wykrywanie anomalii, anomalie sieciowe, wykrywanie ataków

Introduction

Nowadays data integrity and security are required quality standards that must be met. Security in network traffic is one of the basic problems faced by network infrastructure that is exposed to public traffic. In addition, the infrastructure is exposed to attacks of various types and not only is a source of data leakage but also generates costs related to the handling of network traffic generated by the attacker.

The task of outlier detection algorithms is to find such patterns in the searched vectors that are incompatible with the expected characteristics. Outlier detection research involves studies in a wide range of fields and offers a broad spectrum of possible applications, such as monitoring the activities of enemies, detecting traps or unidentified objects, detecting fraud on credit cards and bank transfers, fraudulent transactions, insurance fraud, or detecting hacker attacks. Each group of the above-mentioned applications deals with a different data type. Initially, outliers were treated as noise, or incorrect data.

This article describes how to collect data describing the cluster network traffic, resource consumption and the characteristics of processing unit usage. Cluster, whose main task was to create and verify electronic signatures for PDF documents, will provide us needed data. The respective nodes were loaded both internally and externally to differentiate the data. The collected data were grouped by the algorithm proposed by the authors, using popular grouping methods. The goal of the algorithm was to decide whether a given sample was generated as a result of a network attack or as normal traffic. A positive result of the experiment were the basis for the automation of the entire process and the creation of a complete algorithm for detecting anomalies in network traffic.

The paper is structured as follows. Section II gives a short overview of the literature, with a particular emphasis on the works concerning outlier detection. Section III presents the basic definitions of an outlier and an anomaly in the context of network traffic parameters and the load on the cluster system resources, which stand out from the standard readings. Next, in Section IV, selected outlier detection methods applied to collected data are briefly discussed. Finally Section was presented results and conclusion.

Related works

The related literature is divided into articles relating to the methods used in this article for anomaly detection and articles defining the basic concepts and serving as a statistical comparison of existing solutions for anomaly detection. In the category of papers describing methods, [1] presents a rather outdated method for anomaly detection that was the first to approach the problem of anomaly detection in a gradual way, instead of binary. The method presented in [2] is a clustering algorithm for density-based data classification that requires only one input parameter. [7] is an improvement of [1], based on nearest neighbor distance. The novel anomaly detection method presented in the [4] article is a response to extensive unsupervised methods. It describes a simple to build and implement algorithm that achieves better results than existing solutions. A detailed description of the methods is presented in the Methods section. The article [5] describes the use of SVM (Support Vector Machine) and GA (Genetic Algorithm) algorithms for anomaly detection. It also presents current available tools for monitoring network traffic in real time. [3] focuses on distinguishing between attacks and normal network traffic. Attacks are divided into 4 main categories (Dos, Probe, R2L, U2R).

The basic definitions

Before we start talking about anomalies, we should understand what they are. We can define an exception very generally as a pattern of characteristics that deviates from the expected "normal" behavior of other objects in the analyzed data set. General definitions of an exception are given e.g. Aggrawal 2013, Hawkings etc. An exception can therefore be defined in the following form:

Definicja 1 *The general definition of the exception*
Let a set of X objects be given. Each $x_i \in X$ object is described by the feature vector $x_i = \{a_1, a_2, ..., a_n\}$. The x_w object defined by the feature vector $x_{w1}, x_{w2}, ..., x_{wn}$ is called an exception, or a singular point or an outlier if the feature vector of x_w significantly differs from the feature vectors of the remaining elements (objects) of the set X.

The purpose of this paper is to identify deviations from the operating norm of a cluster that supports network traffic of implemented applications. An anomaly in the context of this work will be all readings of network traffic parameters or the load on the cluster system resources, which stand out from the standard readings. Therefore, anomalies are all deviations from the standard traffic. This traffic is generally characterized by periods of increased intensity, during peak user-traffic hours, such as office hours, and by periods

of much lower activity, such as night hours. In the context of such traffic, an unexpected increase in load may constitute both a potential attack attempt and an anomaly.

Depending on the way the cluster is loaded, we can distinguish several types of attacks. In the case of an application that supports HTTP requests, the potential attack generated by testing by the authors is an HTTP flood. This is a simple-to-operate, non-spoofing attack. It involves loading the endpoint with a number of queries beyond the capabilities of the attacked system. Due to the complex structure, HTTP queries are much more overwhelming for the system that supports requests than simpler floods of TCP or UDP type, and at the same time, they are not so burdensome for the attacker. It generates both the network load as well as a load of hardware resources. Due to the specifics of the simulated document signing system, a potential attack may be overloading the system with large files, requiring more time to calculate the hash and taking up more disk space. Inherent in systems where external files are accepted, there is a risk of attacks using specially prepared files that can remotely access the server part of the infrastructure and gain unauthorized access to the data within it.

Methods

For initial anomaly detection analysis, the DBSCAN[2], LOF[1] and COF[7] methods were used, as they are well known in the literature for use in this kind of problem and helped verifying the usability of data collected from the cluster. PCA[6] and ECOD[4] methods were also used and compared with methods mentioned above.

The DBSCAN (*Density Based Spatial Clustering of Applications with Noise*) [2] method groups together densely placed objects. Objects in the lower density spaces are considered exceptions. For each object, a neighborhood is defined that consists of objects less than eps away from it. If there are more than n objects in the object's neighborhood, it is marked as a core object and all points adjacent to it belong to the same group as that core object. Objects that do not belong to any group are exceptions. For this method, it is very important to select the appropriate value of the eps parameter, which depends on the characteristics of the data set and the arrangement of objects in it. The smaller the value of the eps parameter and the greater the number of n objects, the denser the cluster of objects that makes up the group. Unlike the rest of the methods described in this section, the DBSCAN method does not assign any confidence factor to an object, but rather explicitly determines whether it is an exception or not. For that reason, during the training, multiple eps values need to be tested to find the one that results in the closest contamination level to a given one.

The LOF (*Local Outlier Factor*) [1] method assigns each object an outlier factor based on its local density compared to the local density of its *n* nearest neighbors. An object is an exception if its outlier factor exceeds a certain threshold. The advantage of this method is that it works well on groups with different object distribution densities and – unlike DBSCAN – it does not require specifying the distance at which the vicinity of the object will be examined.

The reachability distance of an object p with respect to object o is defined as:

$$\textit{reach-dist}_k(p,o) = max(d(p,o), \textit{k-dist}(o)), \tag{1}$$

where $d(p,o)$ is the distance between p and o, and *k-dist*(o) is the distance to k-th nearest object of o. $N_k(p)$ is k-neighborhood of p, which is the set of points that are within *k-dist(p)* of p.

Local reachability density of an object p is defined as:

$$lrd_k(p) = \left(\frac{\sum_{o \in N_k(p)} \textit{reach-dist}_k(p,o)}{|N_k(p)|} \right)^{-1}. \tag{2}$$

Local outlier factor is defined as:

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|}. \tag{3}$$

The COF (*Connectivity-based Outlier Factor*) [7] method is very similar to the LOF method. Instead of local density, it examines the "connection" of an object to its neighbors. A set based nearest path (SBN-path) of an object p on set of its k-neighborhood $N_k(p)$ is denoted as $s = \{p_1, p_2, ..., p_r\}$, such that for all $i \in [1; r-1]$ object p_{i+1} is the nearest neighbor of set $\{p_1, p_2, ..., p_i\}$ in set $\{p_{i+1}, p_{i+2}, ..., p_r\}$. An object's distance from a set is defined as its distance from its closest object in the set. A trail of a SBN-path s is a set $e = \{e_1, e_2, ..., e_{r-1}\}$, such that for all $i \in [1; r-1]$ object e_i is a pair $\{o_i, p_{i+1}\}$, where o_i is the nearest object in set $\{p_1, p_2, ..., p_i\}$ of an object p_{i+1}.

The average chaining distance is defined as:

$$\textit{ac-dist}(p) = \sum_{i=1}^{r-1} \frac{2(r-i)}{r(r-1)} d(o_i, p_{i+1}) \tag{4}$$

For a given object p, the Connectivity-based Outlier Factor is defined as:

$$COF_k(p) = \frac{|N_k(p)| \textit{ac-dist}(p)}{\sum_{o \in N_k(p)} \textit{ac-dist}(o)} \tag{5}$$

The ECOD (*Empirical-Cumulative-distribution-based Outlier Detection)*) [4] method treats anomalies as rare, unlikely events. It is assumed that the feature distributions are independent of each other. The left and right tail ECDFs (*empirical cumulative distribution functions*) for the j-th feature are defined as:

$$F^{(j)}_{\textit{left}}(z) = \frac{1}{n} \sum_{i=1}^{n} b(X_i^{(j)} \leq z) \tag{6}$$

$$F^{(j)}_{\textit{right}}(z) = \frac{1}{n} \sum_{i=1}^{n} b(X_i^{(j)} \geq z) \tag{7}$$

where b is a function that evaluates to 1 for a true expression and 0 otherwise. $X_i^{(j)}$ is a value of a j-th feature of an i-th object.

The outlier scores can be defined as:

$$O_{\textit{left-only}}(X_i) = -\sum_{j=1}^{d} log(F^{(j)}_{\textit{left}}(X_i^{(j)})) \tag{8}$$

$$O_{\textit{right-only}}(X_i) = -\sum_{j=1}^{d} log(F^{(j)}_{\textit{right}}(X_i^{(j)})) \tag{9}$$

$$O_{\textit{auto}}(X_i) = -\sum_{j=1}^{d} (b(\gamma_j < 0) log(F^{(j)}_{\textit{left}}(X_i^{(j)})) + b(\gamma_j \geq 0) log(F^{(j)}_{\textit{right}}(X_i^{(j)}))), \tag{10}$$

where:

$$\gamma_j = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i^{(j)} - \overline{X^{(j)}})^3}{(\frac{1}{n-1}\sum_{i=1}^{n}(X_i^{(j)} - \overline{X^{(j)}})^2)^{3/2}} \tag{11}$$

Final outlier score is:

$$O_i = \max\{O_{\textit{left-only}}(X_i), O_{\textit{right-only}}(X_i), O_{\textit{auto}}(X_i)\} \tag{12}$$

The PCA method (*Principal Component Analysis*) [6] transforms objects to an eigenvector space and computes their outlier scores based on the weighted sum of distance from origin. First a correlation matrix is computed from a matrix of n rows of observations and k columns of random variables. If $(\lambda_1, e_1), (\lambda_2, e_2), ..., (\lambda_k, e_k)$ are the k eigenvalue-eigenvector pairs of said correlation matrix, such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k > 0$, then the i-th principal component of an object p is:

$$y_i = e_i z = e_{i1} z_1 + e_{i2} z_2 + ... + e_{ik} z_k, \tag{13}$$

where $z_i = \frac{p_i - \overline{p_i}}{\sqrt{s_{ii}}}$, where $\overline{p_i}$ and s_{ii} are the sample mean and the sample variance of the variable X_i. The outlier score is equal:

$$\sum_{i=1}^{k} \frac{y_i^2}{\lambda_i} = \frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} + ... \frac{y_k^2}{\lambda_k} \tag{14}$$

Results

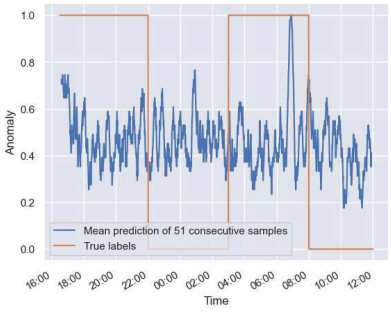
The network traffic was generated by a script created for this article. At the beginning, a script generating a simulation of the traffic of real users was run. PDF files were sent to the application with random frequency in order to generate their digital signature. During the simulation of normal traffic, attacks were performed - the script sent files for signing at the maximum frequency, which made the cluster overloaded. Network traffic and other data was collected using the Prometheus cite Prometheus tool. A total of 24 traits were selected to analyze the collected data. Before the analysis, each of the features was normalized using the formula:

$$z = (x - \overline{x})/\sigma. \tag{15}$$

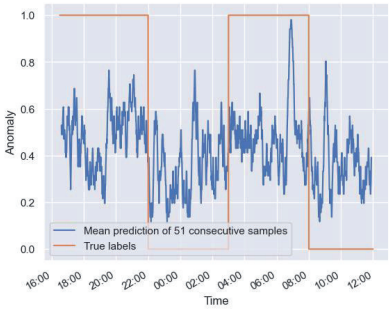
x is a normalized feature, $sigma$ is the standard deviation and $overlinex$ is the mean.
The set generated by us has been divided into a training set and a test set. The training set was about 20 hours of samples and the test set was about 10 hours. Samples were taken every 15 seconds. The graphs Fig.1 and Fig.2 show the average prediction of 51 consecutive samples, compared to the graph showing the actual attacks. The table 1 shows each quality measure based on known attack times.

Method	Set	Quality measures			
		Accuracy	Precision	Recall	f1-score
LOF	Training	0.51	0.55	0.45	0.49
	Test	0.51	0.51	0.56	0.53
COF	Training	0.51	0.55	0.48	0.51
	Test	0.47	0.47	0.45	0.46
ECOD	Training	0.69	0.74	0.64	0.69
	Test	0.71	0.70	0.73	0.72
PCA	Training	0.76	**0.83**	0.71	0.76
	Test	**0.79**	0.78	**0.80**	**0.79**
DBSCAN	Training	0.56	0.61	0.48	0.54
	Test	0.57	0.56	0.59	0.58

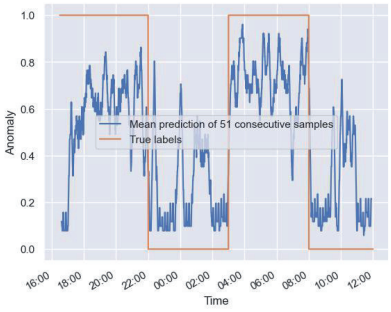
Table 1. The results of individual methods on the test and training set based on the known times of attacks - anomalies.



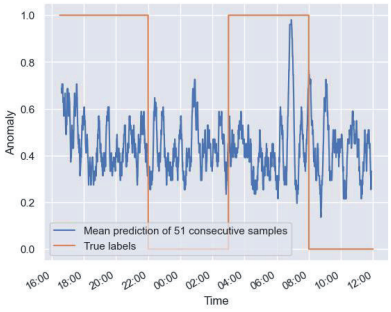
(a)



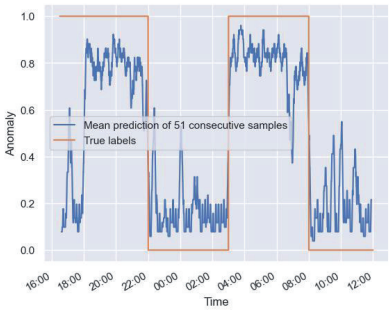
(b)



(c)

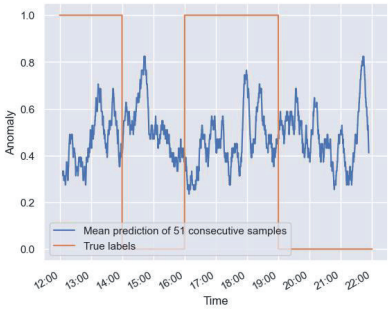


(d)

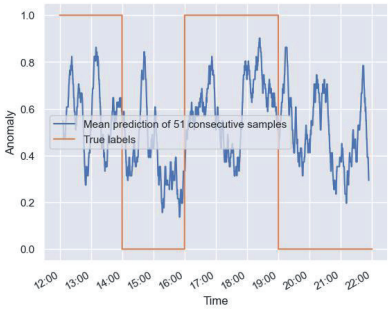


(e)

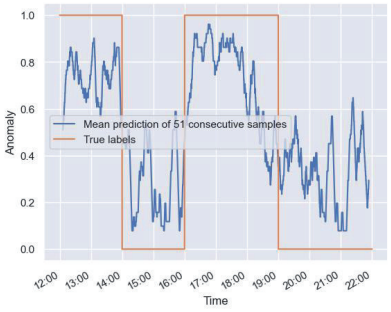
Fig. 1. Graphs showing the mean prediction of 51 consecutive samples juxtaposed with the graph showing actual attacks for the training set. 1 is an anomaly, 0 is normal traffic. Methods: a) COF, b) DBSCAN, c) ECOD, d) LOF, e) PCA



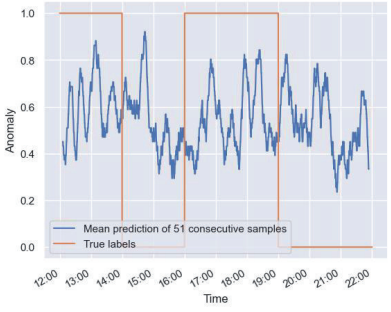
(a)



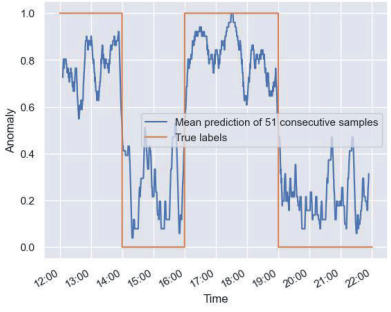
(b)



(c)



(d)



(e)

Fig. 2. Graphs showing the average prediction of 51 consecutive samples juxtaposed with a graph of actual attacks for the test set. 1 is an anomaly, 0 is normal traffic. Methods: a) COF, b) DBSCAN, c) ECOD, d) LOF, e) PCA

Summary

The generated traffic made it possible to clearly define the effectiveness of the selected methods. The PCA method coped best with the task, the accuracy and precision of which for the test set remained similar to the results of the training set n, and additionally obtained results oscillating at the level of 0.8. The second method showing a satisfactory precision was the ECOD method, the results of which were 0.7. Contrary to expectations, the LOF, COF and DBSCAN methods fared worse than the more advanced methods. The results were less than 0.6 which borders on the randomness of the results. Due to their characteristics, these methods had trouble distinguishing noise from generated load, so the results were inconclusive. The more advanced methods, PCA and ECOD, did better due to the reduction of dimensionality, respectively, allowing the elimination of less significant features and the treatment of features individually.

The research allowed to determine which methods will allow for reliable, automated in the future detection of anomalies and prevention of their occurrence. These methods, additionally, thanks to supervised science, allow for adjusting the operation to the cluster specification, therefore they are universal and applicable to other infrastructure.

*

References

[1] Markus M Breunig et al. “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[2] Martin Ester et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[3] Félix Iglesias Vázquez and Tanja Zseby. “Analysis of network traffic features for anomaly detection”. In: *Machine Learning* 101 (Dec. 2014). DOI: 10.1007/s10994-014-5473-9.

[4] Zheng Li et al. “Ecod: Unsupervised outlier detection using empirical cumulative distribution functions”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[5] Taeshik Shon et al. “A machine learning framework for network anomaly detection using SVM and GA”. In: *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*. 2005, pp. 176–183. DOI: 10.1109/IAW.2005.1495950.

[6] Mei-Ling Shyu et al. *A novel anomaly detection scheme based on principal component classifier*. Tech. rep. Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering, 2003.

[7] Jian Tang et al. “Enhancing effectiveness of outlier detections for low density patterns”. In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2002, pp. 535–548.