ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Anomaly Detection Approach in Cyber Security for User and Entity Behavior Analytics System

Vladimir Muliukha, Alexey Lukashin, Lev Utkin, Mikhail Popov, and Anna Meldo *

Peter the Great St. Petersburg Polytechnic University - Research Laboratory of Neural Network Technologies and Artificial Intelligence
Saint-Petersburg - Russia

Abstract. This paper presents a prototype of an intelligent system for advanced analytics for integrated security of complex information and cyberphysical systems with the implementation of analytical models and software developed in Peter the Great St. Petersburg Polytechnic University. The article discusses the practical aspects of the application of unsupervised machine learning methods to the tasks of identifying abnormal objects in the field of information security in computer networks. The format of presenting initial data on various events in computer networks is described, as well as the process of preparing a training set for machine learning. The results of detecting anomalies by the Isolation Forest and Local Outlier Factor methods are presented, as well as an analysis of the results.

1 Introduction

Currently, in corporate networks there is a need to protect information resources from unauthorized access and fraud, both by users of corporate networks and by intruders. For the entire volume of stored data, it is necessary to ensure security and to prevent leaks. Therefore, there is a need to detect attempts of unauthorized access in real time.

Corporate networks mainly use Security information and event management (SIEM) solutions related to the User Behavior Analytics (UBA) category. These systems allow analyzing users' behavior. Each action or sequence of users' actions in the corporate network is checked for compliance with security policy. If a rule is violated, a specific action described by security analysts is done. For example, if a user of a corporate network for 5 times in a row in 2 minutes entered the incorrect password while logging in, then access to the authorization page is blocked for him for some time. The main problem of this approach is that the security policy is static, and the capabilities and methods of cyber-attacks are constantly changing.

In 2015, Gartner Company in a study "Market Guide for User and Entity Behavior Analytics" described a new category of UEBA, which in addition to analyzing user's behavior includes analysis of devices, applications, servers, routers and other active "entities" of network interaction. In order to switch from UBA to UEBA in corporate networks, it is necessary to implement a system capable of processing information about the network status in real time, as well as information about the actions of users and "entities" within the network and automatically detect anomalies using primarily machine learning methods.

* The reported study was funded by RFBR, project number 19-29-01004.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

The detection of abnormal behavior of systems based on machine learning methods is a well-studied area [1],[2],[3]. Many works use the Mahalanobis distance between objects as an indicator of abnormality. It is calculated by the formula:

$$d(x, y) = \sqrt{(x - y)^T S^{-1}(x - y)},$$

where x and y are n-dimensional vectors, and S is the covariance matrix. The Mahalanobis distance is a generalization of the concept of Euclidean distance. It differs from the Euclidean distance in that it takes into account correlations between variables and precisely because of this it is scale invariant. If the covariance matrix S is the identity matrix, then the Mahalanobis distance will be equal to the Euclidean distance. Using this metric imposes certain restrictions on the data. The main and most important limitation is that the number of examples should be no less than the dimension of the vectors, i.e., no less than the number of signs. The main drawback of the method for detecting anomalies based on the Mahalanobis distance is that the Mahalanobis distance cannot be calculated when the dimension of the variables exceeds the number of observations. In addition, it mainly considers the linear relationship between vectors (elements of the training set). One of the approaches, which can be attributed in a sense to the nonlinear analogue of the Mahalanobis distance, is the Siamese neural network, which consists of two identical networks with common parameters [4],[5],[6]. At the input of the Siamese network, vectors are given that describe a pair of objects, and at the output, a distance is determined based on the semantic proximity of pairs of objects. It is quite simply proved that in the case of linear network activation functions, the distance at the output of the network in the simplest case corresponds to the Mahalanobis distance. However, when using nonlinear activation functions and with a specific determination of the semantic proximity of pairs of objects, one can obtain more complex analogues of the Mahalanobis distance and analyze the anomalous behavior of the systems. In fact, Siamese networks are also a tool in the construction of robust metric distance models (distance metric learning) [7],[8],[9], in accordance with which, distances between the elements of the training sample are transformed to increase the efficiency of classifiers, which is done by grouping data of one class. At the same time, for many applied tasks, an effective tool for detecting abnormal behavior is various compositions of Siamese networks and autocoders, which have hardly been offered in the literature to date.

2 Practical Implementation of Machine Learning for Anomaly Detection

During this project we've developed and created a prototype of a system for detection of cybersecurity incidents in intensive stream of heterogeneous semi-structured events from different agents/sensors in a corporate computer network. To implement the prototype in real time with an incoming flow of 10,000 eps (events per second), a 10 core server with 64 GB of RAM was required.

The application has a service-oriented architecture and is divided into the following subsystems:

1. Data gathering subsystem, which is designed to receive messages about events

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

from different sources (currently SIEM) and store them in the storage, as well as accumulating a window (sequence of events) and transmitting it for operational analysis. To gain access to the stored data, the system provides an API that allows you to transfer messages based on specified criteria to other subsystems;
2. Data processing subsystem, which is designed to identify anomalies in the data and manage the life cycle of data analysis subsystems (anomalies) and trained models. Also, the subsystem contains the functionality of storing sets of labeled or reference data and training models using them;
3. User interface for analytics and control which is designed to work with the system, set up settings and other necessary operations by working through the web interface.

2.1 Data preparation

The detection of anomalies involves the detection of objects that are not similar to most objects in the training set, i.e., distinguished objects. Moreover, most often in the training set there are either absolutely no anomalous objects, or there are very few of them and it is not known which objects are anomalies. For this reason, the problem of detecting anomalies can be approached by unsupervised learning methods. As a result, the solution of the problem is reduced to determining how much the new object is similar to the objects from the training set. Based on the similarity and its established border, a decision is made whether the new object is an anomaly or not.

To analyze the state and activity of a corporate network, it is necessary to aggregate and analyze the state and activity of all the “entities” in the network, whether it is a user, service, server, switch, etc. Each device for each event writes some information about it to the log. ArcSight introduced the CEF format in 2006 for all event messages. In the CEF format, besides the rest, there is a required Extension field that is a set of key-value pairs. A set of predefined keys contains more than one hundred keys. Because the use of these keys in the Extensions field is not necessary and these keys did not occur in all events, in this work, we analyzed the data to identify the most significant keys. The analysis included a study on the frequency occurrence of keys within one type or category of devices and a study of the semantic component of the keys, based on the documentation provided. Thus, only a small part of these keys was used, which either met in a large number of events and carried quite an understandable meaning, or met in a small number of events, but carried an understandable meaning and a certain importance even with a small amount.

In our project we’ve used 8 keys: “Proto”, which identifies the protocol used in transport layer 4, “Src”, which identifies the source of the event by IPv4 address, “Dst”, which identifies the destination address of the event by IPv4 address, “SourceZoneURI”, which identifies the URI of the source zone of the event, “DestinationZoneURI”, which identifies the URI of the destination zone of the event, “Spt” that is a port of the event source, “Dpt” that is the destination port of the event, and “BytesIn” that is the number of bytes transmitted from the source to the destination during the event.

Further, for each key signs were selected. For example, having analyzed the values of the Src key, the following data were obtained, which are presented in Table 1. Each value corresponds to the number of 1 million events.

After analyzing the obtained data, it was decided to use only two values as

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

signs: the number of events with the Src key value from the subnet 192.*.*.* and the number of events from all other subnets because number of these events is large enough.

Total number of events	Number of events with Src	Total different IP	Different subnets X.Y.Z.*	Different subnets X.Y.*.*	Different subnets X.*.*.*	Events from subnet 192.*.*.*	Events from other subnets
1000000	571279	5258	3246	2180	187	511488	59791

Table 1: Data obtained after analysis of Src key values.

Thus, a separation is made into the sources of events from the internal network and the external one. A similar separation of attributes was made with the values of all other keys. Thus, 40 signs were obtained. In addition to the fields corresponding to the CEF format, each event also contained a timestamp field, the value of which corresponded to the time of the event in milliseconds. After data analysis and feature extraction, data were aggregated with a given time interval for further use in machine learning methods and algorithms.

2.2 Machine learning methods

The methods chosen to detect anomalies at this stage of the project are the following: Isolation Forest and Local Outlier Factor. These methods were chosen in connection with various approaches to data processing. For the first method, with the described "random" method of constructing trees, emissions will fall into the leaves at early stages (at a shallow depth of the tree), i.e., emissions are easier to "isolate". The second postulates the existence of certain metrics in the space of objects, which helps to find anomalies. Because the Local Outlier Factor method measures the distance between objects, it is important to choose a suitable metric.

The analyzed data are events that occurred in one business day in the investigated organization. The number of events is 19 million. The time of the first event is 8:00 (8 a.m.) The time of the last is 22:00 (10 p.m.) Thus, the time period in which all events are located is 14 hours. A 5-minute interval was chosen to describe the state of the system, which made it possible to obtain 168 data vectors. As a result, it was found that the number of vectors is sufficient to calculate the covariance matrix. At various time intervals, the number of events in the corporate network ranged from 39 764 to 546 651, and the average number of events in each vector was 115 583.

All 19 million events are aggregated with a given time interval of 5 minutes. After that, each group of events is converted into a numerical vector of features, each of which describes the state of the corporate network for a specific period of time. Next, the resulting list of vectors is transmitted to the input of each of two methods. At the output of each method there is a list of anomalies discovered by it.

For the Isolation Forest method, 100 trees in the ensemble were selected. This value was chosen with a margin based on the convergence of the path length. The number of trees is determined manually based on the number of features and the

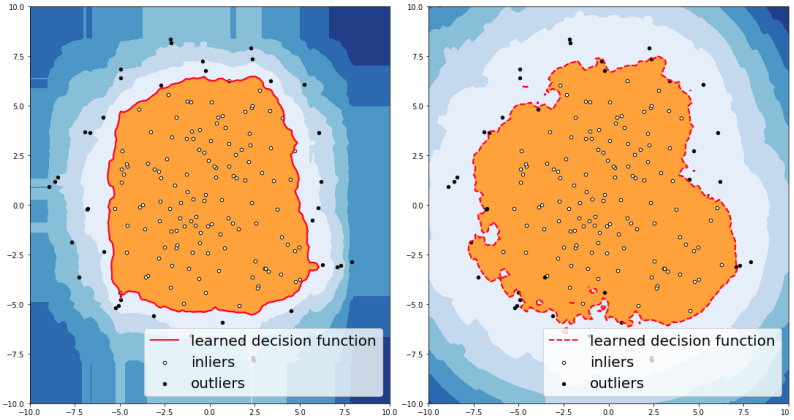
ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

number of training samples. This is a custom parameter, selected to improve the quality of the model.

For the Local Outlier Factor method, the value of the neighbors with which the local density is compared is selected. It is a configurable parameter, but the optimal value is considered equal to the square root of the number of training samples. In this case, the value was set to 13, since 168 data vectors were processed.

Also, for the method, it is necessary to indicate the boundary value with respect to which a decision will be made whether the object is an anomaly or not. As mentioned earlier, there are no specific rules for choosing this value and its selection is based on a priori considerations for the probability of an anomaly occurrence. After a certain launches of method with different values, a number 0.25 was chosen.

For a visualization of the obtained results, it is necessary to switch from 40-dimensional space to 2-dimensional space. This transition was done using the t-SNE method. The t-SNE algorithm finds a two-dimensional representation of the data that preserves the distances between objects in the best way. The t-SNE method could use the Mahalanobis distance as a metric. Thus, using this method, data was converted into 2-dimensional space without losing the initial distance parameter.



(a) (b)

Fig. 1: Visualization of the results obtained and the decisive function of the Isolation Forest method (a) and Local Outlier Factor method (b).

Analyzing the obtained visualizations, the differences in approaches to detecting anomalies in methods are clearly visible in Fig.1. The decisive function of the Isolation Forest method, in contrast to the Local Outlier Factor, is less curved, because when forming trees, it strictly divides the space into subspaces with respect to a randomly selected axis. In turn, the crucial function of the Local Outlier Factor is very close to the boundary anomalous objects, and the boundaries of the crucial function of the Isolation Forest are located approximately in the middle between the anomalous and normal objects. This is due to the fact that the Local Outlier Factor at each point in space has a specific local density value obtained by rough construction of the method model, and Isolation Forest constructs the model by constructing a set of models each of which is obtained by randomly dividing the space into subspaces,

thus after averaging a sufficiently large number of randomly constructed models, a more averaged solution is obtained. While comparing these methods by the degree of retraining, the Local Outlier Factor model turns out to be more retrained than Isolation Forest due to a more rough construction of the model.

3 Conclusion

An anomaly detection approach in cyber security for UEBA System was presented. A prototype of an intelligent system for advanced analytics in information security was developed. Initial data was presented as 19 million events in CEF format. For the machine learning methods 8 keys with 40 signs were chosen. Chosen ones either met in a large number of events or carried an importance even in a small number of events. Two unsupervised methods were chosen to detect anomalies: Isolation Forest and Local Outlier Factor. For the first method, emissions in the data set are easier to "isolate" during the tree formation. In the second method the Mahalanobis distance between objects is formed to find anomalies. Isolation Forest was chosen as a preferable one for the proposed task due to a more complex crucial function.

References

[1] A. Lukashin, M. Popov, A. Bolshakov and Y. Nikolashin, Scalable Data Processing Approach and Anomaly Detection Method for User and Entity Behavior Analytics Platform, *Intelligent Distributed Computing XIII*, pages 344-349, Springer, 2020.

[2] S. Erfani, S. Rajasegarar, S. Karunasekera and C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, *Pattern Recognition*, vol.58, pages 121-134, Elsevier, 2016.

[3] L.V. Utkin, A framework for imprecise robust one-class classification models, *International Journal of Machine Learning and Cybernetics*, vol. 5(3). pages 379-393, Springer, 2014.

[4] S. Berlemont, G. Lefebvre, S. Duffner and C. Garcia, Siamese neural network based similarity metric for inertial gesture classification and rejection, In proceedings of *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition* (FG), 1, pages 1-6, IEEE, 2015.

[5] L. Bertinetto, J. Valmadre, J. Henriques, A. Vedaldi and P. Torr, Fully-Convolutional Siamese Networks for Object Tracking. In proceedings of the *14th European Conference on Computer Vision* (ECCV 2016: Workshop) Computer Vision – ECCV 2016 Workshops 9914, pages 850-865, Springer, 2016.

[6] G. Koch, R. Zemel and R. Salakhutdinov, Siamese neural networks for one-shot image recognition. In proceedings of the *32nd International Conference on Machine Learning*, vol. 37, pages 1-8, Lille (France), 2015.

[7] J. Hu, J. Lu and Y.-P. Tan, Discriminative deep metric learning for face verification in the wild. In proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2014), pages 1875-1882, IEEE, 2014.

[8] C. Li, M. Georgiopoulos and G. Anagnostopoulos, Kernel-based distance metric learning in the output space. In proceedings of the *2013 International Joint Conference on Neural Networks* (IJCNN 2013), pages 1-8, IEEE, 2013.

[9] X. Yin and Q. Chen, Deep Metric Learning Autoencoder for Nonlinear Temporal Alignment of Human Motion. In proceedings of *IEEE International Conference on Robotics and Automation* (ICRA 2016), pages 2160-2166, IEEE, 2016.