State of the Art Literature Review on Network Anomaly Detection

Tero Bodström(✉) and Timo Hämäläinen

Faculty of Information Technology, University of Jyväskylä, Agora, P.O. Box 35, 40014 Jyväskylä, Finland
tero.bodstrom@gmail.com,timo.hamalainen@jyu.fi

Abstract. As network attacks are evolving along with extreme growth in the amount of data that is present in networks, there is a significant need for faster and more effective anomaly detection methods. Even though current systems perform well when identifying known attacks, previously unknown attacks are still difficult to identify under occurrence. To emphasize, attacks that might have more than one ongoing attack vectors in one network at the same time, or also known as APT (Advanced Persistent Threat) attack, may be hardly notable since it masquerades itself as legitimate traffic. Furthermore, with the help of hiding functionality, this type of attack can even hide in a network for years. Additionally, the expected number of connected devices as well as the fast-paced development caused by the Internet of Things, raises huge risks in cyber security that must be dealt with accordingly. When considering all above-mentioned reasons, there is no doubt that there is plenty of room for more advanced methods in network anomaly detection hence more advanced statistical methods and machine learning based techniques have been proposed recently in detecting anomalies. The papers reviewed showed that different methods vary greatly in their performance to detect anomalies. Every method had its advantages and disadvantages, however most of the presented methods cannot detect previously unknown attacks but on the contrary, for example, detects DDoS attacks extremely well.

Keywords: Network attacks · Anomaly detection · Machine learning

1 Introduction

The purpose of this study is to highlight major challenges in network anomaly detection using statistics and machine learning, excluding deep learning, by focusing on recent research in the field. Among the growing number of data and network connected devices, the challenge is different attack types such as APT, DDoS and Zero-day. They each have a unique behavioural pattern and the difficulty is to come up with a solution that has the capability to detect all of them efficiently in modern networks. In this study following aspects are

O. Galinina et al. (Eds.): NEW2AN 2018/ruSMART 2018, LNCS 11118, pp. 89–101, 2018.
https://doi.org/10.1007/978-3-030-01168-0_9

considered: intended attack type detection, functional differences as well as the differences in detection accuracy.

By definition, anomalies are observations which differ from other observations enough to arise suspicion. Suspicious observations in network traffic can be caused by either legitimate events or non legitimate events and the purpose of anomaly detection is to divide normal and anomalous data with different techniques [1,2]. However, any suspicious event has to be treated as hostile, until it is verified and proved to be non-hostile. Most of the presented studies in this paper are focused on DDoS, Zero-day and web attacks. There is less current research material on APT attacks for some unknown reason and one focus in this paper is to evaluate the possibility for using presented methods to detect APT attacks.

This paper unfolds as follows: in the second section research papers based on different anomaly detection technologies are presented. The third section summarizes perceived improvements for the presented researches. The fourth section concludes this review discussing advantages and disadvantages of selected research presented along the paper.

2 Network Anomaly Detection

In this section different methods that use machine learning for network anomaly detection is presented. These methods focus on one or more simultaneous attack types.

The number of DDoS attacks are increasing due to the growing number of IoT devices with low security mechanisms and the fact that nowadays it is fairly easy to acquire attack tools. This has created a situation where large number of these devices can be used to perform distributed attacks, that is, to carry out more powerful attacks [3–7]. So far 24 different DDoS attack vectors have been found globally [3]. There exist researches that focuses on detecting these types of attacks specifically, to be used as a first line defence.

2.1 A Lightweight Network Anomaly Detection Technique

To serve as a frontier in a network anomaly detection system, Jinoh Kim et al. present in their research a new lightweight grid-based approximation technique. Their proposed method is based on a recursive algorithm that partitions data in D-dimensional space. In each recursion, the algorithm verifies if data belongs to a sub-block of a grid and if yes, the data is labelled and the execution of the algorithm stops. Otherwise, the algorithm continues to the next sub-block and execution continues until the data gets labelled. In case the algorithm ends without labelling, the data gets classified as "*Not Sure*". In their experimental tests, they used only two variables to train the system and detect data: src and dst bytes. The former is the number of bytes from the source to the destination IP addresses, and the latter is the number of bytes from the destination to source hosts, including five continuous attributes (duration, src bytes, dst bytes, wrong segment and urgent) in the group. This method used six different classes

for classification: "*Secure*", "*Marginal Secure*", "*Marginal Insecure*", "*Insecure*", "*Empty*", and "*Not Sure*". The authors also made comparison tests with more traditional methods such as decision tree and random forest. Tests for the proposed method showed the following results, (i) accuracy of 98.5% with the KDD data and (ii) 83% with NSL-KDD. In addition, the measured learning time for their method is significantly lower and approximately two orders of magnitude faster than decision tree and random forest [4].

The proposed method detected DDoS attacks successfully and the detection speed was fast. On the other hand, it would be interesting to see if these features will keep up when adding variables for more precise detection.

2.2 Distributed, Multi-level Network Anomaly Detection for Datacentre Networks

Mircea Iordache et al. presents in their research a distributed method for network anomaly detection. In the proposed method, all network devices are equipped with a detection algorithm and they contribute to detecting anomalies. The algorithm has consensus voting mechanism, where at first, each node independently detects if an anomaly exists and then peer nodes vote for the final decision. In addition, the devices are capable of creating attack path reconstruction for further investigation purposes. The actual algorithm has two detection layers to gain better accuracy. The first level performs a coarse-grain detection for fast analysis. In case the algorithm detects a potential anomaly, data is passed to the second, fine-grained detection level. The deeper sub level performs analysis and collects data metrics. The authors selected sketch-based algorithm for fine-grained detection. For in-depth analysis, they modified the Count-Min Algorithm. The purpose was to store source and destination IP addresses and transport layer ports to sketch but this approach enabled also storing the time variant state of the flow and transferring it to the sketch. Therefore, any changes in flow can be detected by comparing network data to stored metrics and used to detect presence of anomalies [5].

The authors tested the proposed method for various attack types, such as Brute Force access, 0-day attacks and Port Scans. Their test results concluded that the solution was able to offer complete path reconstruction at the onset of DDoS attacks that generally have high intensity. Also, partial path reconstruction was achieved for anomalies of lower intensity [5].

For future work M. Iordache et al. mentioned an improvement for synchronizing remotely invoked instances by advanced majority voting scheme. The purpose of the improvement is to extend the amount of classification information what will be used for anomaly detection. Secondly, they proposed further research in improving method deployment and management techniques by taking into account (i) data locality, (ii) system resource availability and demand and (iii) guaranteeing network topology changes seamlessly [5].

The proposed method was intended for Data centres, but it can be used also in a local networks. When comparing to grid-based approximation, this solution is more comprehensive as it also offers also full path reconstruction and can

detect other type of attacks. However, the accuracy rate is lower and this type of solution does not fit well for APT attack detection as it has extremely low intensity when hiding in a network [8].

2.3 Distributed Network Anomaly Detection on an Event Processing Framework

Atanas Pamukchiev et al. research focused on distributed event process called Network Intrusion Detection System (NIDS) for data centres. Still today, approaches for NIDS are expensive to implement and they cover only segments of network focusing on data streams passing through fixed points. Another challenge in data centres internal networks is high speed, which can reach 40/100GbE. Distributed NIDS aims to reduce cost and also increase detection performance in these complex high speed networks that provides a large and varying numbers of services, such as cloud servers, data instances, data storage, image and facial recognition services [9].

Their proposed system architecture relies on topology of Apache Storm, Directed Acyclic Graph (DAG), which is used for distribution functionality. Apache Storm functionality is mapped directly to data center network topology. This is possible since Bolt nodes in Storm are similar to switches and Spout nodes are similar to core routers as well as to hosts in a data center. Similar topologies allows direct mapping regardless of the complexity of network. For bidirectional detection in a network, two identical topologies are needed due to one direction restriction in a Storm and DAG [9].

The authors proposed a lightweight Storm module directly integrated to fabric switch to facilitate network implementation. Their system detection modules are responsible for extracting data from packets and perform detection independently. Data extraction can be executed by fields or network layers and an aggregated method detects more complex anomalies. Though the proposed system is distributed, it requires a centralized management and configuration node. In all distributed nodes an API is required for runtime configuration. Controller node also monitor Bolts; it keeps track and check their state and behaviour. Controller also ensures quick recovery when problems occur [9].

Test results for the prototype of proposed system are as follows. According to the authors first tests were executed with various anomalies, such as invalid fields, blocked source and destination addresses, application layer packet contents marked as anomalous and even Denial of Service attacks. The first tests presented 20% contamination with 7,32% decrease in system throughput. The second test executed with 50% contamination and system throughput dropped 7,32%. All packets were processed in less than 7 ms [9].

Installing detection modules directly to network elements, for example in a switch, is one possible solution to resolve problems in a complex network. Tests demonstrated that integration decreases overall throughput when anomalies increase. In normal data center usage, these changes are not visible to end-users but when a solution requires almost real-time response time, even a latency of 7 ms starts to matter. The proposed system does not map directly to deep

learning methods, as algorithm requires more calculation power than a fabric switch has. However, adding a lightweight deep learning module to the switch, might solve the problem.

2.4 Big Data Analytics for Network Anomaly Detection from Netflow Data

Duygu S. Terzi et al. proposed in their paper an anomaly detection method based on big data analytics from NetFlow data. Aim was to detect anomalies caused by UDP flood from specific IPs. Proposed method collects network, users, applications and routing traffic and uses clustering based unsupervised machine-learning for detection. They used six steps in their method, (i) Interval division in Netflows, (ii) Source IP Aggregation in Netflows, (iii) Standardization of data using z-score, (iv) aggregated NetFlows were clustered by K-mean algorithm as distributed, (v) Calculation of Euclidian distance between the cluster center and elements, (vi) Determination of normal and abnormal flow numbers from time intervals used in steps iv and v [6].

They implemented Netflow with Apache Spark Cluster and Azure HDInsight with python program. They reduced 6-dimensional data to 3-dimensions to help with accuracy and visualization using a statistical method called principal component analysis (PCA) [6].

The proposed method detected anomalies with 96% accuracy. The authors tested the proposed method with labelled botnet CTU-13 dataset. Data set includes 13 different scenarios and for the study, they selected the 10th due to size of dataset and number of attacks. "*The data set has 4.75 h records and 1309791 flows covering 106352 UDP DDoS flows*" [6].

Terzi et al. mentioned that most of the network traffic is normal, and due to this, anomalies and outliers based on network attacks are rare and this causes negative effect on anomaly detection research and development. For future work they proposed gathering more network data to gain better detection results as well as innovative algorithms and platforms [6].

The proposed method detected 96% of UDP DDoS attacks. Authors used PCA to reduce data dimensions from six to three, however, the above-mentioned reduction in dimensions caused some normal data to be detected as an anomaly. Using Support vector clustering (SVC) the possibility for false positive detection from occurring could be reduced. SVC maps data points from data space to high dimensional feature space with kernel function so dimension reduction is not needed, that is, all possible data is available for detection process.

2.5 Entropy-Based Network Anomaly Detection

Christian Callegari et al. proposed anomaly-based IDS system, where the detection is executed with a sketch algorithm for estimating the quantity of entropy in the data stream. The authors state that estimation of the entropy associated to the traffic descriptors has shown to be promising choice in anomaly detection. The proposed method is based on estimating different kinds of entropy. Their

study includes following sections: (i) three-dimensional reversible sketch and (ii) combined sketches with entropy estimation, (iii) implementation of different kinds of entropy and (iv) impact of different traffic detection for system performance. For system and detection tests they used MAWILAB traffic recordings. They focused on finite discrete distribution and comparison of two empirical distributions. For entropy measures, the authors used "*some kind of entropy*". That is there exists two possible ways to do comparison: (i) comparing entropies of two distributions or (ii) checking relative entropy between distributions [10].

The Authors used NetFlow and Flow-Tools module Data Formatting for processing data, which was collected from router during fifteen minute time-bins. After the data was processed and formatted correctly, it was used as an input to sketch algorithm for constructing reversible sketch tables. Instead of two-dimensional array, sketch tables used three-dimensional data structures and the histograms were stored in third dimension. Random aggregation was added to the algorithm to avoid mimicry attack. That is, without randomization the attacker could try to mimic an actual attack and that would create a situation, where the histograms has exactly the same values. The constructed sketch tables was used for anomaly detection tests [10].

During the tests a scatter plot was created with two variables, Byte and Flow. Both variables were tested with five different entropy methods, (i) Shannon entropy, (ii) Tsallis entropy, (iii) Renyi entropy, (iv) Kullback-Leibler divergence and (v) Jensen-Shannon divergence. Test results show that different detection accuracies can be achieved by varying the entropy method used. Best detection accuracy, over 85%, resulted in a Flow test using Tsallis' method. Byte testing demonstrated good results with Shannon's and Thallis' methods as well as with Jensen-Shannon divergence. Test results were decent with known attacks, but unfortunately the proposed system was unable to detect unknown attacks so there was no improvement when comparing to the traditional IDS [10].

The proposed system has good detection performance, but tests were executed only with recorded data. It would be interesting to execute performance tests in a real-time environment and verify how the proposed system performs and what is the detection latency, if any.

2.6 Combination of R1-PCA and Median LDA for Anomaly Network Detection

Elkhadir et al. proposed a method with two combined dimension reduction algorithms for anomaly detection. Selected algorithms were Rotational Invariant L1-norm Principal Component Analysis (R1-PCA) and median Linear Discriminant Analysis (median LDA), and the focus was on detecting anomalies of Denial-Of-Service and Network Probe attacks [7].

The authors stated that the origin of PCA comes from minimizing the sum of squared errors and it is very sensitive for outliers. In their proposed method rotational invariance was used instead, which searches for the principal eigenvectors of a covariance matrix. Thus, R1-PCA has a significant drawback, as it may give more weight to features with higher variability disregarding their

effectiveness. However, LDA searches first a projection matrix and then uses a class median vector to estimate a class mean vector. LDA has a known small sample size problem, that is, when the actual data has more dimensions than the training data, computing becomes impossible [7].

The authors used KDDcup99 dataset for testing the proposed method and they converted all discrete values of the dataset to continuous values. For accuracy testing two factors were used: (i) detection rate and (ii) false positive rate. Test results show that their proposed method was able to identify 95,5% of DoS attacks and 86,7% of Probe attacks, at highest. DoS attack detection was 94,7% and probe detection 71,6%, at lowest. Test result table shows that there was no linear dependency compared with training data and detection rate [7].

As future work Elkhadir et al. stated that they would like to test the proposed dimension reduction algorithms with multidimensional network with real multidimensional network data including images and text [7].

The proposed method is well suited for detecting DoS attacks, even with low number of training data. For detecting Probe attacks, the amount of training data has to be higher, as tests showed that lowest detection rate was 71,6% and that probably is not enough for good detection rate. Since tests were executed only with KDDcup99 dataset, it would be interesting to know what was the detection latency and also how well this method performs in a real environment. Moreover, due to the few drawbacks in the method that the authors presented, malware may be able to manipulate TCP packets in real environment which leads to non-standard data causing difficulties in detecting anomalies.

2.7 Integrating Short History for Improving Clustering Based Network Traffic Anomaly Detection

Juliette Dromard et al. proposed in their paper an unsupervised network anomaly detector. The proposed method's aim was to detect anomalies without prior knowledge or attack fingerprints caused by zero day attacks. The authors mentioned few most studied methods, such as K-means, SVM, DBSCAN which relies on clustering and PCA. Current detectors do not consider temporal information. Due to temporal existence of attack data and the mentioned lack of methods, they focused on studying H-ORUNDA(History Online Real-time Unsupervised Network Anomaly detection Algorithm), which is an improved version of ORUNDA algorithm. The algorithm was modified to keep temporal record of clustering results and it was implemented on Spark Streaming big data platform for reducing detection latency [11].

In the first phase, the authors define three rare incidents which may contain important and interesting historical data: (i) data flow that was similar to other data flows in the past, but has been modified since, (ii) data flow which statistic change suddenly and (iii) data flow which appear or disappear suddenly. Three new parameters were defined in the second phase: (i) length of history in seconds, (ii) threshold, if number of points of a cluster change, it can be considered as an anomaly and (iii) threshold d, if a point moves at least d distance, it is considered as a flow change. To improve the algorithm, 15 second-time slots

were chosen since it gave good results in terms of true positive and false positive rates. Time-slots are then divided into micro-slots, which improves real-time detection. Collected data has to be aggregated to different flow levels and they chose seven aggregation levels. Every aggregated level has a unique flow matrix and every flow is a set of features and is stored in aggregation level matrix. The detector process every matrix independently. The authors also mentioned about the curse of dimensionality phenomena, what happens with high dimensions. "*In high dimensions, distance becomes meaningless and every point tends to become an outlier*". To avoid this problem, they used subspace clustering and evidence assembly, which divided the entire space into subspaces and partitioned every subspace independently. To speed up the detection process, they also implemented incremental grid clustering algorithm (IGDCA) which can discover any shapes of cluster and it identifies noise too [11].

For the evaluation tests, SynthONTS dataset was selected. It is a real world sanitized data traffic gathered by a Spanish operator and it contains lot of different type real anomalies and artificially injected anomalies. For detection testing they made two different type of tests, (i) detection performance and (ii) detection time. For comparison methods PCA and DBSCAN were chosen. The tests results showed following: H-ORUNDA has high detection rate and low false positive rate and it performed better than methods compared in general. The performance test for detection time showed that reduction of micro-slot improved average runtime and proposed method can process incoming traffic faster than it arrives while the size of a micro-slot is at least 0,3 s. The authors also tested proposed method in Google Cloud Platform and its purpose was to test hardware requirements and scalability for real-time detection. Spark Streaming did not perform satisfactorily and they mentioned that "*a simple parallel implementation in C on a simple PC performs better*" [11].

The study had a few really interesting approaches, such as time-slotting flow and implementing IGDCA for any cluster shape. Time-slotting could help also in APT attack detection as those are really discreet and try to use slow traffic as a masquerade, among the other methods. Also multiple cluster shapes without restrictions can help to detect these type discreet attacks. Their proposed data flow aggregation may be worthy for further studies.

2.8 Network Anomaly Detection Based on Dynamic Hierarchical Clustering of Cross Domain Data

Yang Liu et al. proposed Dynamic Hierarchical Clustering of Cross Domain Data based anomaly detection in their paper. They focused on improving real-time detection and existing clustering methods, which are sensitive and easily fall to local optimal solution. The authors also mentioned that it is difficult to achieve real-time detection with machine learning and deep learning methods [12].

The proposed method uses cross domain hybrid data for anomaly detection. Hybrid data contains both, categorical and numerical data, and was added to unified framework instead of analysed separately. Euclidean distance was extended with frequency information to add dynamic clustering accuracy, that

is, to measure similarity of cluster centres and samples. Their algorithm used the following execution sequence: (i) set dynamic clustering accuracy to evaluate accuracy of clustering process, (ii) execute new cluster analysis for classes that does meet accuracy requirements, (iii) repeat ii as long as is required and (iv) finally, tree clustering structure is trained by ongoing hierarchical clustering with training data. For defining clustering accuracy, disturbance in cluster class was used as an indicator. This accuracy determined if cluster needed K- means algorithm to execute second step, hierarchical cluster analysis [12].

KDDCup99 10% dataset was used for testing the proposed method and it contains different type of network attacks and intrusion behaviour. Also, the test set had 17 types of attacks which were not included in the training set. However, KDDCup99 10% dataset contains a lot of duplicates, which can cause that different types of attacks are added to same cluster leading to lower clustering performance. For comparison tests they selected following algorithms: (i) basic K-means, (ii) improved K-means, (iii) AGFCM and (iv) Naive Bayes. Test results show that the proposed method had highest detection rate (98.2%) and lowest false detection rate (5.72%] in a comparison test [12].

With the proposed method the authors were able to achieve good detection results, including low false detection rate. On the other hand, the method's performance in a real-time detection was not tested while it was mentioned that some other approaches are not suitable for real-time detection. It would be interesting to see how the proposed method performs in a real-time environment.

2.9 Probabilistic Transition-Based Approach for Detecting Application-Layer DDoS Attacks in Encrypted Software-Defined Networks

Elena Ivannikov et al. proposed a method for detecting Application-Layer DDoS Attacks in Encrypted Software-Defined Networks (SDNs), with Probabilistic Transition-Based Approach, that is, an algorithm which extracts statistics directly from data flows in SDN and compare behavioural patterns to normal traffic in order to detect significant anomalies. Research focus for cloud environments were defined because of the rapid growth in business use, thus it can be comprehended as a critical part of modern business and DDoS can cause huge problems for companies and their reputations by disabling services. Cloud environments can adapt quickly to fluctuating demands and SDN has eased network maintenance and configuration tasks [13]. However, it has created a situation where cloud networks and attacks are increasing in complexity every day [9,13]. The main difference between traditional and application layer DDoS attacks is that while traditional DDoS attacks are executed at the network layer, the application layer attacks are executed at the seventh OSI layer and it tries to mimic legitimate traffic to avoid detection. HTTP is commonly used in application layer DDoS attacks and it targets the vulnerabilities in operative systems and web applications [13].

The authors proposed a method where packet headers were extracted from data for detection purposes, which also enables using encrypted traffic. To start

with, the authors built a model called *conversation* for normal user behaviour which was a collection of short time sequence data. They implemented four ways to characterize each *conversation*, (i) source IP, (ii) source port, (iii) destination IP or iv) destination port. Besides characterization the following information was extracted: (i) duration of *conversation*, (ii) number of packets sent in 1 second, iii) number of bytes sent in one second, (iv) average packet size and (v) presence of TPC flags such as URG, ACK, PSH, RST, SYN and FIN. For the detection method they selected two clustering algorithms, (i) k-means and (ii) Clustering Using REpresentatives (CURE), and then calculated probabilistic transition for clusters [13].

For executing tests, first they applied clustering algorithms to extracted features, which created representations of distinct groups and discovered hidden patterns in data traffic. In the second phase *conversations* were grouped together based on characterization and time interval. In the third phase each session was represented in every time window based on cluster sequence from the first step. Last phase was to estimate conditional and marginal probabilities for every sequence and calculate threshold. To solve a real time detection problem for previously unknown behavioural patterns of users, they used streaming k-means algorithm which allows updates to the behavioural data in the trained model. A so-called forgetting mechanism was also implemented to the update mechanism where after time the old models get less important for the actual updated model. Detection tests were executed with trained user behavioural models and by using normal net bank traffic and intermediate DDoS attack with several bots-attackers trying to mimic regular browsing behaviour. Test results showed that the proposed clustering algorithms with probabilistic transition-based approach *k-means+Prob* performed with the highest accuracy, respectful 99.58% and with false positive rate at zero. The authors also mentioned that the proposed method gained three improvements compared to earlier studies: (i) performance, that is, low false alarm rates and high detection accuracy (ii) reduced number of effective parameters in cluster, only one and (iii) significant reduce for required storing space [13].

For the future work Ivannikov et al. mentioned improving detection accuracy with bigger dataset and focusing on detecting more advanced DDoS attack with simulation [13].

Due to its ability to detect anomalies from the seventh OSI layer with encrypted data, the proposed approach could be used also for APT attacks. However, the forgetting algorithm would be needed to be revised due to the tendency for the APT attacks to hide unnoticed as long as possible to not forget old traces completely before the next *conversation*.

3 Summary of Further Improvements

In this section the perceived improvements for more precise anomaly detection is presented. These identified concerns vary from single to multiple papers.

Even though numerous papers mentioned that KDDCup99 is not realistic, data is obsolete and lack of modern network traffic, it is widely used for benchmarking new methods. The research community should go forward and look for present day datasets or create those to substitute KDDCup99, as in some papers researchers had already done.

In many of the proposed methods, the authors used dimension reduction to fit the multidimensional data to the two-dimensional space. This can cause losses of critical data and lead to lower accuracy rates and increases in false alarm rates.

In some of the papers only certain parts of IP and TCP headers were selected for the anomaly detection, such as source and destination IP's and ports. These types of solutions can limit the detection capability due to the vast amount of the possibly useful missing data.

In one paper, proposed method focus was in UDP DDoS attacks detection, hence the usage of method is highly limited.

In a few of the papers tests were executed only with recorded data and not a single test was executed in a real network environment. Results of these types of tests cannot be used for benchmarking anomaly detection performance in a real situation.

Some of the proposed methods required high intensity attacks to perform the anomaly detection properly and thus the usage is rather limited. In addition, almost every paper where the anomaly detection was executed by traditional machine learning or statistical approach suffered from the same performance shortage, lack of detecting the earlier unknown attacks. This is a common problem for the information security products which rely on signature based detection.

In one paper it was stated that some other approaches are not suitable for real-time detection but interestingly the authors did not perform any testing in real-time environment with their own proposed method.

Baddar et al. pointed out in their paper in 2014, that in several papers they investigated it was assumed that the majority of network traffic is normal. In reality that might be even the opposite, for example in attacks such as DDoS [14]. That assumption was also present in many of the papers which were presented in this review.

4 Conclusion

The papers reviewed showed that different methods vary greatly in their performance to detect anomalies. Every method had its advantages and disadvantages, however most of the presented methods cannot detect previously unknown attacks but on the contrary, for example, detects DDoS attacks extremely well. The methods mentioned were based on statistics and traditional machine learning, while those methods have slight disadvantages, they have advantages also, such as real-time detection and low memory consumption. Current network devices, such as firewall, IDS, switch and so forth, can be complemented with

the proposed methods to gain more visibility to ongoing situation in networks. However, proposed methods does not fit for APT attacks detection due to its sophisticated behaviour and tendency to hide in networks even for years and mimic normal traffic.

When developing anomaly detection systems and methods, these advantages and disadvantages should be further considered, as they can help to define what could the actual focus of the work. With current practices a system or a method that could detect all types of attacks, not to mention in a real-time environment, requires enormous resources and might be still even impossible to implement.

References

1. Andropov, S., Guirik, A., Budko, M., Budko, M.: Network anomaly detection using artificial neural networks. In: 2017 20th Conference of Open Innovations Association (FRUCT) (2017). https://doi.org/10.23919/FRUCT.2017.8071288
2. Aygun, R.C., Yavuz, A.G.: Network anomaly detection with stochastically improved autoencoder based models. In: 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing, pp. 193–198 (2017). https://doi.org/10.1109/CSCloud.2017.39
3. Yuan, X., Li, C., Li, X.: DeepDefense: identifying DDoS attack via deep learning. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP) (2017). https://doi.org/10.1109/SMARTCOMP.2017.7946998
4. Kim, J., Yoo, A., Sim, A., Suh, S., Kim, I.: A lightweight network anomaly detection technique. In: 2017 Workshop on Computing, Networking and Communications (CNC) (2017). https://doi.org/10.1109/ICCNC.2017.7876251
5. Iordache, M., Jouet, S., Marnerides, A.K., Dimitrios, P.P.: Distributed, multi-level network anomaly detection for datacentre networks. In: IEEE ICC 2017 Next Generation Networking and Internet Symposium (2017). https://doi.org/10.1109/ICC.2017.7996569
6. Terzi, D.S., Terzi, R., Sagiroglu, S.: Big data analytics for network anomaly detection from netflow data. In: 2017 International Conference on Computer Science and Engineering (UBMK), pp. 592–597 (2017). https://doi.org/10.1109/UBMK.2017.8093473
7. Elkhadir, Z., Chougdali, K., Benattou, M.: Combination of R1-PCA and median LDA for anomaly network detection. In: Intelligent Systems and Computer Vision (ISCV) (2017). https://doi.org/10.1109/ISACV.2017.8054985
8. Ussath, M. Jaeger, D., Cheng, F., Meinel, C.: Advanced persistent threats: behind the scenes. In: 2016 Annual Conference on Information Science and Systems (CISS) (2016). https://doi.org/10.1109/CISS.2016.7460498
9. Pamukchiev, A., Jouet, S., Pezaros, D.P.: Distributed network anomaly detection on an event processing framework. In: 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 659–664 (2017).https://doi.org/10.1109/CCNC.2017.7983209
10. Callegari, C., Giordano, S., Pagano, M.: Entropy-based network anomaly detection. In: 2017 International Conference on Computing, Networking and Communications (ICNC): Communications and Information Security Symposium (2017). https://doi.org/10.1109/ICCNC.2017.7876150

11. Dromard, J., Owezarski, P.: Integrating short history for improving clustering based network traffic anomaly detection. In: 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W), pp. 227–234 (2017). https://doi.org/10.1109/FAS-W.2017.152
12. Liu, Y., et al.: Network anomaly detection based on dynamic hierarchical clustering of cross domain data. In: 2017 IEEE International Conference on Software Quality, Reliability and Security (Companion Volume), pp. 200–204 (2017). https://doi.org/10.1109/FAS-W.2017.152
13. Ivannikova, E., Zolotukhin, M., Hämäläinen, T.: Probabilistic transition-based approach for detecting application-layer DDoS attacks in encrypted software-defined networks. In: Yan, Z., Molva, R., Mazurczyk, W., Kantola, R. (eds.) NSS 2017. LNCS, vol. 10394, pp. 531–543. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64701-2_40
14. Baddar, S., Merlo, A., Migliardi, M.: Anomaly detection in computer networks: a state-of-the-art review. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. (JoWUA) **5**, 29–64 (2014). https://www.researchgate.net/publication/270274504_Anomaly_Detection_in_Computer_Networks_A_StateoftheArt_Review