Network Anomaly Detection

An Trung Tran
Advisor: Marcel von Maltitz, Stefan Liebald
Seminar Innovative Internet Technologies and Mobile Communications SS2017
Chair of Network Architectures and Services
Departments of Informatics, Technical University of Munich
Email: antrung.tran@tum.de

ABSTRACT

In this paper we present the foundations of Network Anomaly Detection, which include the definition of a Network Anomaly Detection System, its purpose, challenge. This paper also provides an overview over the different types of attributes extractable from raw network data. Another valuable aspect of this paper is the taxonomy of various different algorithm types, which are described, in a concise way. This includes the main advantages and drawbacks of each type and an evaluation of the methods as well as two examples of algorithms.

Keywords

Network Anomaly Detection, Feature Selection, Algorithms, Network Intrusion Detection

1. INTRODUCTION

According to Anderson [2], an intrusion attempt or threat is defined as "deliberate and unauthorized attempt to (i) access information, (ii) manipulate information, or (iii) render a system unreliable or unusable". With the steady advance of network-based computer systems and modern technologies, there is a increasing need of systems capable of detecting network intrusions, which pose a massive security risk. An example would be the massive Distributed Denial of Service (DDoS) attack on the french web hoster OVH in 2016. This was done by an Internet of Things-based botnet [5], which is a network of private computers infected with malicious software and controlled as a group without the owners' knowledge.

In this paper, we explore one of the measures used to counter such issues, namely Network Anomaly Detection (NAD). NAD is defined as identification of events, which differ from an expected pattern, particularly in network data. Henceforth, in the subsequent parts of this paper, we discuss the different features of the NAD as well as the algorithms and methods applied on the system.

2. NETWORK ANOMALY DETECTION

[4] defines anomaly detection in networks as "the problem of finding exceptional patterns in network traffic that do not conform to the expected normal behaviour". The outstanding patterns are mostly referred to as 'anomalies' or 'outliers' in this context. There are three types of anomalies: (i) point, which is an instance of individual data that has been found anomalous, e.g. a purchase with large transaction value; (ii) contextual, a data instance found anomalous in a specific context, e.g. a large spike in traffic in the middle of the night and (iii) collective, a collection of data instances found anomalous, e.g. breaking rhythm in an electrocardiogram (ECG) [4]. Anomalies can be divided into two categories [14]: (a) performance related anomalies and (b) security related anomalies. In this paper, we focus on the security related anomalies which occur due to malicious activities intended to flood the network with unnecessary traffic hijacking the bandwidth and turning the system inaccessible, i.e. a DDoS attack as mentioned before.

2.1 Properties of NADS

Network Anomaly Detection Systems (NADS) serve the main purpose of processing network data by monitoring packets on the network and look for patterns and is used to determine whether the input data is an anomaly or a normal data instance. According to [4], NADS is based on five different characteristics which describe the concept: (i) "Principal assumption: All intrusive activities are necessarily anomalous." (ii) The method compares the normal system state with an established profile. When the degree of deviation is too high, intrusion attempts are reported. (iii) False positives are anomalous activities, which are not intrusive. (iv) "One should select threshold levels so that neither of the above two problems is unreasonably magnified nor the selection of the features to monitor is optimized." (v) "Computationally expensive because of overhead and possibly updating several system profile matrices". Aside from that, NADS operate on three different modes: (i) supervised, which uses both training data from normal and anomaly classes; (ii) semi-supervised, which only use labeled instances of data for the normal classes and (iii) unsupervised, which requires no labeled instances of data but labeling is done by the system itself [4].

2.2 Challenge of NADS

The biggest challenge for Network Anomaly Detection Systems is the definition of the "concept of normality" [4], which is defined "by a formal model that expresses relations among the fundamental variables involved in the system dynamics" [4].

Therefore, instances are detected as anomalous, if the degree of deviation with respect to the normal profile is too high. [4] introduces an abstract model of an anomaly detection system S that uses a supervised approach. Training datasets are used and labelled for normal as well as for the anomaly class. S is defined as $S = (M,D)$ with M being the

doi: 10.2313/NET-2017-09-1_08

normal model and D being "a proximity measure that allows one to compute, given an activity record, the degree of deviation that such activities have with regard to the model M. Each system can be broken down into 2 modules: (i) a modeling module, which trains the system to achieve the normality model D and (ii) a detection module, which uses (i) to classify new traffic as anomalous or outliers. M needs to be flexible in order to handle changing scenarios".

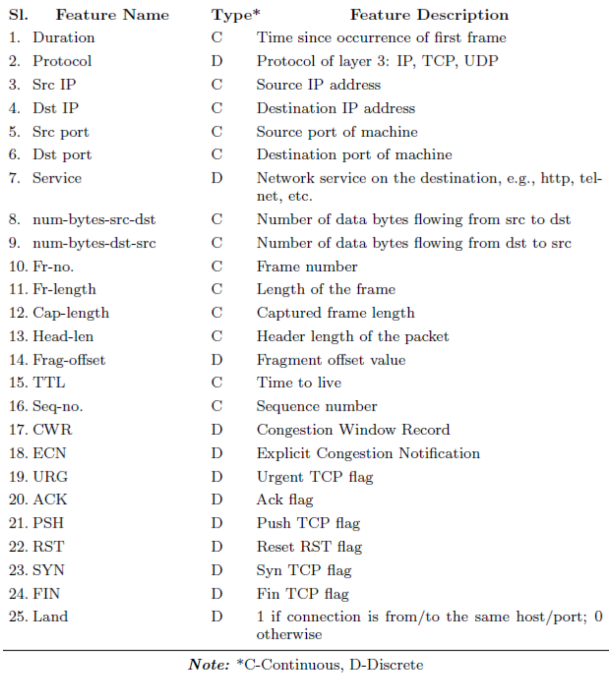
3. FEATURES

Features are defined as attributes extractable from raw network data, in which its selection is crucial for network anomaly detection. Feature selection is the process of extracting specific features out of raw data to be loaded into an algorithm because not all algorithms work with all kinds of data. Most data can be classified into one of two groups : (a) numerical or (b) categorical. Numerical data can be divided into two more subgroups, which are discrete data (representing items which can be counted) and continuous data (representing measurements or values that can only be described using real number intervals). Feature selection offers a lot of advantages because it (i) improves the performance of an algorithm as it cuts down on feature dimensionality , (ii) removes insignificant features , (iii) improves data quality and therefore the efficiency of learning algorithms, (iv) raises the detection rate and (v) helps in understanding the data generation process as well as visualizing it [3] [4].

When looking for useful features, there are various ways to view a connection. First, it can be done by inspecting individual packets and their characteristics stored in the headers. Another method of viewing a connection could be done by analysing the packet flows from the source to destination and vice versa. In the following we are analyzing the TUIDS dataset described in [3], which divides features into three different categories: (i) packet traffic feature dataset, (ii) network flow traffic feature dataset, (iii) portscan. We are only going to analyze (i) and (ii) as the survey does not mention a significant reason to additionally consider (iii). The features listed in this dataset will be compared to other surveys the features and complemented by using other sources.

3.1 Packet Traffic Features

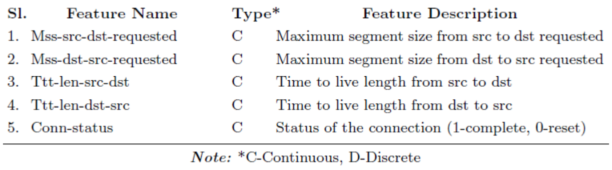
This section describes the possible features which can be extracted by inspecting individual packets and their headers. These are otherwise known as packet traffic features. This information can be used to create statistics to detect anomalies. Other algorithms which utilize this information to detect anomalies are available aswell. The information itself can be extracted by looking at the connection and sampling packets in different timeframes or at different points in time. Sampling is necessary here as perusal of the full packet data would be very time consuming and costly. Packet traffic features can be divided into (i) basic packet features (see Figure 1), (ii) content-based packet features (see Figure 2), (iii) time-based packet features (see Figure 3) and (iv) connection-based packet features (see Figure 4). The Figures 1-4 are giving a general overview of the features which are applicable for later uses. Besides the listed features from [3], there are other features, which can be used, when inspecting into individual packets and their headers. In comparison [11] uses IP packet size as well as TCP Header Size, TCP Window Size and TCP options and some of the listed features. [9] used the S0 Flag, which is the first SYN packet sent, when the 3-Way-Handshake for TCP is established, as well as the rejection (REJ) flag as features and some of the listed features. In [12] identification number is used together, with acknowledgement number and the options.



Sl.	Feature Name	Type*	Feature Description
1.	Duration	C	Time since occurrence of first frame
2.	Protocol	D	Protocol of layer 3: IP, TCP, UDP
3.	Src IP	C	Source IP address
4.	Dst IP	C	Destination IP address
5.	Src port	C	Source port of machine
6.	Dst port	C	Destination port of machine
7.	Service	D	Network service on the destination, e.g., http, telnet, etc.
8.	num-bytes-src-dst	C	Number of data bytes flowing from src to dst
9.	num-bytes-dst-src	C	Number of data bytes flowing from dst to src
10.	Fr-no.	C	Frame number
11.	Fr-length	C	Length of the frame
12.	Cap-length	C	Captured frame length
13.	Head-len	C	Header length of the packet
14.	Frag-offset	D	Fragment offset value
15.	TTL	C	Time to live
16.	Seq-no.	C	Sequence number
17.	CWR	D	Congestion Window Record
18.	ECN	D	Explicit Congestion Notification
19.	URG	D	Urgent TCP flag
20.	ACK	D	Ack flag
21.	PSH	D	Push TCP flag
22.	RST	D	Reset RST flag
23.	SYN	D	Syn TCP flag
24.	FIN	D	Fin TCP flag
25.	Land	D	1 if connection is from/to the same host/port; 0 otherwise

Note: *C-Continuous, D-Discrete

Figure 1: Basic Packet Features from [3]



Sl.	Feature Name	Type*	Feature Description
1.	Mss-src-dst-requested	C	Maximum segment size from src to dst requested
2.	Mss-dst-src-requested	C	Maximum segment size from dst to src requested
3.	Ttt-len-src-dst	C	Time to live length from src to dst
4.	Ttt-len-dst-src	C	Time to live length from dst to src
5.	Conn-status	C	Status of the connection (1-complete, 0-reset)

Note: *C-Continuous, D-Discrete

Figure 2: Content-based Packet Features from [3]

3.2 Network Flow Traffic Features

This section describes the possible features which can be extracted by inspecting the flows between the source and its destination. This aspect is important as it takes a look at a collection of packets, which would allow the identification of patterns or features otherwise unnoticeable on the level of individual packets. This enables the view on stateful connections such as TCP and the TCP 3-Way-Handshake [9]. Network flow traffic features (Figure 5) are divided into (i) basic features, (ii) time-window features and (iii) connection-based features. In comparison [8] additionally counts the number of packets, acknowledgement packets, retransmitted packets and pushed packets.

Sl.	Feature Name	Type*	Feature Description
1.	count-fr-dst	C	Number of frames received by unique dst in the last T sec from the same src
2.	count-fr-src	C	Number of frames received by unique src in the last T sec to the same dst
3.	count-serv-src	C	Number of frames from the src to the same dst port in the last T secs
4.	count-serv-dst	C	Number of frames from dst to the same src port in the last T secs
5.	num-pushed-src-dst	C	Number of pushed packets flowing from src to dst
6.	num-pushed-dst-src	C	Number of pushed packets flowing from dst to src
7.	num-SYN-FIN-src-dst	C	Number of SYN/FIN packets flowing from src to dst
8.	num-SYN-FIN-dst-src	C	Number of SYN/FIN packets flowing from dst to src
9.	num-FIN-src-dst	C	Number of FIN packets flowing from src to dst
10.	num-FIN-dst-src	C	Number of FIN packets flowing from dst to src

Note: *C-Continuous, D-Discrete

Figure 3: Time-based Packet Features from [3]

Sl.	Feature Name	Type*	Feature Description
1.	count-dst-conn	C	Number of frames to unique dst in the last N packets from the same src
2.	count-src-conn	C	Number of frames from unique src in the last N packets to the same dst
3.	count-serv-src-conn	C	Number of frames from the src to the same dst port in the last N packets
4.	count-serv-dst-conn	C	Number of frames from the dst to the same src port in the last N packets
5.	num-packets-src-dst	C	Number of packets flowing from src to dst
6.	num-packets-dst-src	C	Number of packets flowing from dst to src
7.	num-acks-src-dst	C	Number of ack packets flowing from src to dst
8.	num-acks-dst-src	C	Number of ack packets flowing from dst to src
9.	num-retransmit-src-dst	C	Number of retransmitted packets flowing from src to dst
10.	num-retransmit-dst-src	C	Number of retransmitted packets flowing from dst to src

Note: *C-Continuous, D-Discrete

Figure 4: Connection-based Packet Features from [3]

Sl.	Feature Name	Type*	Feature Description
	Basic features		
1.	Duration	C	Length of the flow (in sec)
2.	Protocol-type	D	Type of protocols– TCP, UDP, ICMP
3.	src IP	C	Src node IP address
4.	dst IP	C	Destination IP address
5.	src port	C	Source port
6.	dst port	C	Destination port
7.	ToS	D	Type of service
8.	URG	D	Urgent flag of TCP header
9.	ACK	D	Ack flag
10.	PSH	D	Push flag
11.	RST	D	Reset flag
12.	SYN	D	SYN flag
13.	FIN	D	FIN flag
14.	Source byte	C	Number of data bytes transferred from src IP addrs to dst IP addrs
15.	dst byte	C	Number of data bytes transferred from dst IP addrs to src IP addrs
16.	Land	D	1 if connection is from/to the same host/port; 0 otherwise
	Time-window features		
17.	count-dst	C	Number of flows to unique dst IP addr inside the network in the last T secs from the same src
18.	count-src	C	Number of flows from unique src IP addr inside the network in the last T secs to the same dst
19.	count-serv-src	C	Number of flows from the src IP to the same dst port in the last T secs
20.	count-serv-dst	C	Number of flows to the dst IP using the same src port in the last T secs
	Connection-based features		
21.	count-dst-conn	C	Number of flows to unique dst IP addrs in the last N flows from the same src
22.	count-src-conn	C	Number of flows from unique src IP addrs in the last N flows to the same dst
23.	count-serv-src-conn	C	Number of flows from the src IP addrs to the same dst port in the last N flows
24.	count-serv-dst-conn	C	Number of flows to the dst IP addrs to the same src port in the last N flows

Note: *C-Continuous, D-Discrete

Figure 5: Network Flow Traffic Features from [3]

4. ALGORITHMS

This section presents various algorithms for NADS, categorizes them for an overview and shows two algorithms as examples. Algorithms in this specific context are mandatory as NAD is not possible without them. There are different approaches by various algorithms to solve the problem of NADS. In the end, the choice of the algorithm will influence the type and quality of the result significantly.

4.1 Classification of NAD Methods

In this subsection the different methods of Network Anomaly Detection are classified and represented as a taxonomy. An algorithm implementing classification is a classifier. In [4] are divided into 6 major categories: (i) statistical-based, (ii) classification-based, (iii) clustering and outlier-based, (iv) soft computing, (v) knowledge-based and (vi) combination learner. These 6 categories are going to be explained in detail in the following subsections in consideration of their advantages and drawbacks.

4.1.1 Statistical-based NAD

For statistical-based NAD, "an anomaly is an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed" [4]. Thus, instances with a low probability of being generated are anomalies. The techniques are divided into (i) parametric and (ii) non-parametric. Parametric techniques "assume knowledge of the underlying distribution and estimate the parameters from the given data" [4], while non-parametric techniques does not. The advantage of these techniques are that they do not require "prior knowledge about normal activity" [6] and can provide accurate notification of malicious activities [4] [6]. The drawbacks are that they are vulnerable to be trained by attackers until "the network traffic generated during the attack is considered normal" [4] and setting values for different parameters and metrics is difficult, especially balancing between false positives and negatives [4].

4.1.2 Classification-based NAD

Classification-based NAD tries to assign new data instances into categories, based on training datasets [4]. Each object can be described using attributes or features. "Linear classification tries to find a line between the classes" [4], but the "classification boundary may be non-linear" too [4] as seen in Figure 6. Advantages of these techniques are that they are capable of improving their execution by integrating new data. Thus, they are adaptable for "training and testing" purposes [4]. Also these techniques have a high detection quota for known anomalies subject to suitable thresholds [4]. The drawbacks are that they are highly susceptible to the hypotheses made by classifiers. Furthermore, they are incapable of detecting unidentified anomalies until applicable training datasets are provided [4].

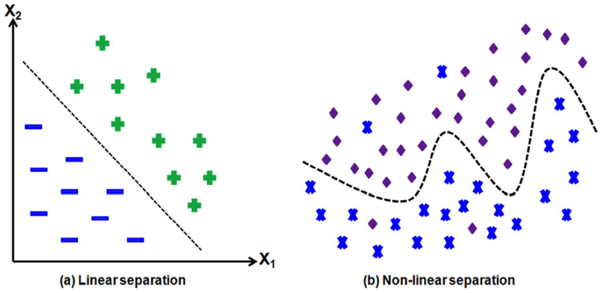


Figure 6: Classification-based NAD from [4]

4.1.3 Clustering and outlier-based NAD

Clustering is grouping new sets of objects into groups called clusters by using a given correlation or measuring distance [6]. Objects in the same cluster are more related to each other than those in other clusters. The most common practice "consists in selecting a representative point for each cluster" [6]. This can be visualised in Figure 7(a), which shows a set of unidentified objects in two dimensions grouped into five clusters by drawing ellipses around them [4]. In Figure 7(b), we then see the separation of outliers (anomalous data points) from the normal clusters, in which these outliers are points which do not fall into any of the clusters formed [4]. The advantages of these techniques are that it is easy to find outliers when working with small-scaled datasets. Another advantage is that "bursty and isolated" [4] anomalies can be analyzed accurately. The drawbacks of these techniques include the fact that only continuous attributes (Examples listed in Chapter 3 Figures 1-5) are used for most of the proposed techniques. In NAD, an hypothesis is that "larger clusters are normal" [4] while smaller ones are attack or intrusions. It is challenging to evaluate these techniques without this hypothesis. The use of disproportionate measurements influences the detection quota negatively. The computational complexity can be quite high when compared to other NAD techniques as most techniques use both clustering and outlier detection [4].

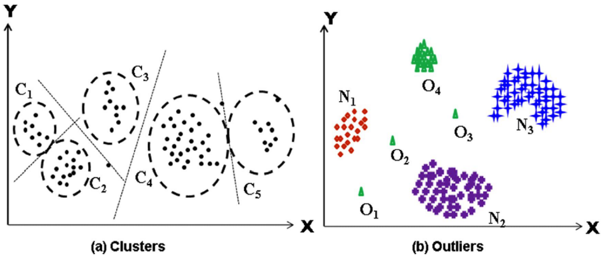


Figure 7: Clustering and outlier-based NAD from [4]

4.1.4 Soft Computing

Soft computing techniques are sufficient for NAD because sometimes it is impossible to find exact solutions [4]. Soft computing is divided into the following methods : (i) Genetic algorithm (GA), (ii) Artificial Neural Networks, (iii) Fuzzy Sets, (iv) Rough Sets and (v) Ant Colony algorithms and artificial immune systems [4].

(i) Genetic Algorithm (GA) is a population-based adaptive heuristic search for finding solutions to optimization and search problems based on the concept of biological evolution [4]. The approach is to evolve a population of possible solutions towards a better solution, which will in turn, result in better detection rates of anomalies. GA uses three main steps to determine the next generation: (a) selection, (b) crossover and (c) mutation. Each solution has a set of properties called chromosomes, which can be altered. Initially, selection involves using a fitness-based mechanism, where a sample of fitter solutions get picked from a huge population of random genes. The higher the fitness score a chromosome has, the better the chance it has to be selected. The selected solutions are then used for crossover. This is when a new type of chromosome is generated by exchanging random selected segments from the chosen chromosomes. The mutation alters existing chromosomes as the new type will now contain parts that cannot be found in the original ones. The process is then repeated until the best suitable solution is found. This solution must meet the criteria that was decided upon before the process and it will be optimal, such that successive iterations will not be improving the results. However, there are some limitations to this method, including the amount of time it may consume for the process [12] [7]. For example [7] uses GA to develop and improve rules for NAD. This example is further explained in Chapter 4.3.

(ii) Artificial Neural Networks (ANN) are inspired by recognition that our brain works in a completely different way compared computers. The brain performs certain computations (e.g. pattern recognition, motor control) much than the fastest computer. ANN is based on a collection of connected units, called neurons. The connection between two neurons is called a synapse and can transmit a signal to another neuron. Thus, in order to achieve a good performance, real neural networks utilize massive interconnections of neurons. Neural Networks learn from their environment by changing interconnection strenghts and synapse weights of the network [4] [6]. In the context of NAD, ANNs are used for data clustering, feature extraction and similarity detection, which is further detailed in [3].

(iii) Fuzzy network intrusion systems are used to determine whether malicious activity is taking place on a network using fuzzy rules. The system combines simple network traffic metrics with fuzzy rules to determine the probability of specific or general network anomalies. Once metrics are available, they are evaluated using a fuzzy set theoretic approach [4].

(iv) Rough set is a mathematical tool for feature extraction in a dataset generating explainable rules for intrusion detection [1].It is used when we do not have complete knowledge of the system. The mathematical framework of rough set theory enables modelling of relationships with a minimum set number of rules. For example, this method first extracts a minimum set of detection rules as the system generates sequences from the normal behaviour model during the execution of a process. With these rules, it then detects any abnormal operating status of the process and reports the abnormality as a possible intrusion. The benefits of using this technique are: (i) enables learning with small size training datasets and (ii) overall simplicity [3] [4] .

(v) Ant colony optimization (ACO) and related algorithms are techniques for solving computational problems, which can be rephrased to search for optimal paths through graphs. The algorithms tries to mimic the behavior of ants searching for a path between their colony and a source of food. Artificial Immune Systems (AIS) represent a computational method inspired by the principles of the human immune system. The human immune system is proficient at performing anomaly detection [4]. ACO in this context is used for feature selection and network anomaly detection, which is further explained in [3].

The advantages of soft computing-based anomaly detection are that these systems show a very high amount of flexibility and adaptability [6]. Unsupervised learning, using competitive neural networks, is effective in data clustering, feature extraction and similarity detection. Drawbacks of these techniques are that they have a high resource consumption as well as high dependency on the hypothesis, about the behaviour accepted for the system. This means that the training of the systems become very difficult, if there is a lack of normal traffic data.[6]

4.1.5 Knowledge-based NAD

Knowledge-based methods are utilizing network or host events and check these against known attack patterns and predefined rule sets. [4] separates these methods into two different categories: (a) rule-based and expert system approaches and (b) ontology and logic-based approaches.

(i) The expert system is a rule-based system, without or with knowledge. This system matches rules against the current state of the system utilizing a rule engine. Depending on the outcome, it fires one or more rules [4]. A very popular example of a rule-based Network Intrusion Detection System is Snort [13], which is an open source intrusion prevention system capable of real-time traffic analysis and packet logging.

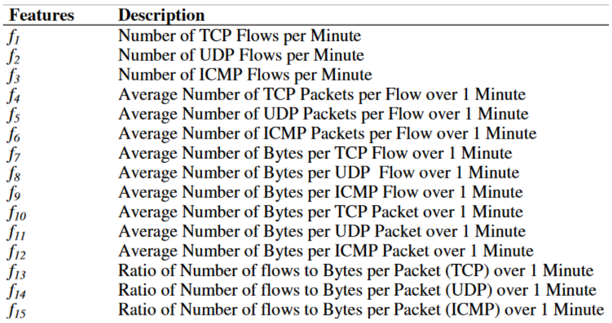
(ii) Ontology and logic-based approaches use expressive logic structure in real time to model attack signatures by integrating statistical properties and constrains [4].

The advantages of these techniques are flexibility, robustness and scalability. Additionally they have a high detection quota, if training datasets are available for both normal state and anomalies. Drawbacks are the costs and time consumption for the development of a high-quality knowledge. It is very challenging for these techniques to detect unknown anomalies [4].

4.1.6 Combination learners

Combination learners are combining multiple techniques and split into three different categories: (i) Ensemble-based methods, (ii) Fusion-based methods and (iii) Hybrid methods [4].

(i) The idea between ensemble-based methods is to consider various classifiers and combining them into one, which outperforms all of these. These techniques evaluate individually and combine them to reach a final verdict. Advantages of ensemble-based methods are that even though the individual classifiers are weak, the ensemble techniques still perform well by combining various classifiers. Another advantage is that they are usable on large-scaled datasets and the set of controlling parameters are extensive and can be easily adjusted. The drawbacks of these techniques are finding consistent performing classifiers from a pool of classifiers is challenging [4].



Features	Description
f_1	Number of TCP Flows per Minute
f_2	Number of UDP Flows per Minute
f_3	Number of ICMP Flows per Minute
f_4	Average Number of TCP Packets per Flow over 1 Minute
f_5	Average Number of UDP Packets per Flow over 1 Minute
f_6	Average Number of ICMP Packets per Flow over 1 Minute
f_7	Average Number of Bytes per TCP Flow over 1 Minute
f_8	Average Number of Bytes per UDP Flow over 1 Minute
f_9	Average Number of Bytes per ICMP Flow over 1 Minute
f_{10}	Average Number of Bytes per TCP Packet over 1 Minute
f_{11}	Average Number of Bytes per UDP Packet over 1 Minute
f_{12}	Average Number of Bytes per ICMP Packet over 1 Minute
f_{13}	Ratio of Number of flows to Bytes per Packet (TCP) over 1 Minute
f_{14}	Ratio of Number of flows to Bytes per Packet (UDP) over 1 Minute
f_{15}	Ratio of Number of flows to Bytes per Packet (ICMP) over 1 Minute

Figure 8: Flow-based features used in [10]

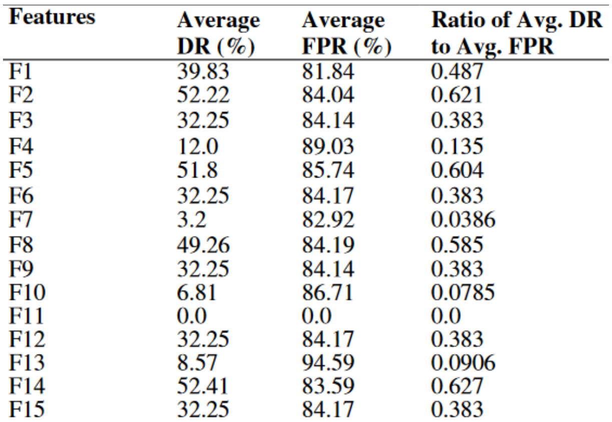
(ii) Fusion-based methods is trying to improve classification accuracy compared to individual decision-based techniques by merging data on (a) data level, (b) feature level and (c) decision level. Data fusion is efficient in terms of increasing timeliness of attack identification and helps reducing false alarm rates. Another advantage would be that decision level fusion have a high detection rate when given applicable training data. The downsides of these techniques are the high consumption of resources as well as the feature level fusion being a time consuming task [4].

(iii) Hybrid methods in [4] are combining various known methods trying to create a new system due to anomaly-based NIDS having a too high false positive rate as well as misuse-based NIDS, which only detects known intrusions. This approach differs from (i) as it also uses misuse-based NIDS. These methods benefit using features from both signature and anomaly-based network anomaly detection. This leads to being able to handle both known and unknown anomalies. The drawbacks of these techniques are that they cost a lot of resources and updating rules or profiles or signatures dynamically remain difficult [4].

4.2 NAD using CUSUM and EM Clustering

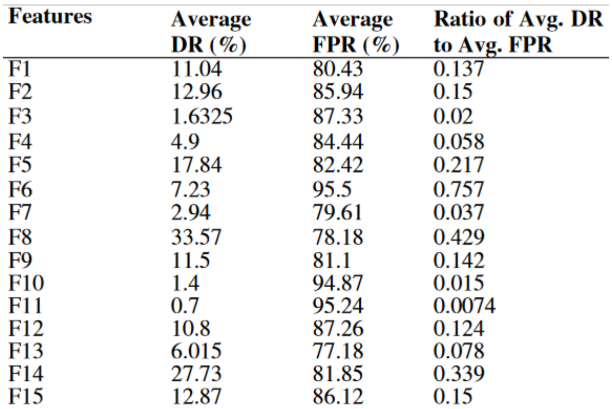
The following section shows an example of a system, which uses CUSUM, non-parametric CUmulative SUM, and EM, Expectation-Maximization, based clustering algorithm [10]. The features used in the survey are all flow-based packet features, which mainly are the number of flows, the average number of packets, bytes per flow and the average number of bytes per packet in a set time interval (Figure 8).

The following Figures 9 (EM) and 10 (CUSUM) show that EM outperforms CUSUM in almost every feature category, as the detection rate (DR) is significantly higher for nearly all features. Although, both techniques show a significant amount of false positive rate (FPR). This leads to the conclusion that despite being able to detect anomalies correctly with a decent rate, the system is buried by false alarms as the quota is the FPR for each feature is almost over 80 percent for each feature in both tests.



Features	Average DR (%)	Average FPR (%)	Ratio of Avg. DR to Avg. FPR
F1	39.83	81.84	0.487
F2	52.22	84.04	0.621
F3	32.25	84.14	0.383
F4	12.0	89.03	0.135
F5	51.8	85.74	0.604
F6	32.25	84.17	0.383
F7	3.2	82.92	0.0386
F8	49.26	84.19	0.585
F9	32.25	84.14	0.383
F10	6.81	86.71	0.0785
F11	0.0	0.0	0.0
F12	32.25	84.17	0.383
F13	8.57	94.59	0.0906
F14	52.41	83.59	0.627
F15	32.25	84.17	0.383

Figure 9: Performance of EM detector used in [10]



Features	Average DR (%)	Average FPR (%)	Ratio of Avg. DR to Avg. FPR
F1	11.04	80.43	0.137
F2	12.96	85.94	0.15
F3	1.6325	87.33	0.02
F4	4.9	84.44	0.058
F5	17.84	82.42	0.217
F6	7.23	95.5	0.757
F7	2.94	79.61	0.037
F8	33.57	78.18	0.429
F9	11.5	81.1	0.142
F10	1.4	94.87	0.015
F11	0.7	95.24	0.0074
F12	10.8	87.26	0.124
F13	6.015	77.18	0.078
F14	27.73	81.85	0.339
F15	12.87	86.12	0.15

Figure 10: Performance of CUSUM detector used in [10]

4.3 Rule-based NAD using GA

Table 1: Training data test results used in [7]

Type	Occurence	Correct Identif.	Incorrect Identif.	Reliability
Normal	5000	4460	540	89.2
Attack	5139	4864	275	94.64

Table 2: Test data test results used in [7]

Type	Occurence	Correct Identif.	Incorrect Identif.	Reliability
Normal	5040	4710	330	93.45
Attack	4958	4670	288	94.19

Table 3: Iterations relative to accuracy in [7]

S #	Iterations	Accuracy (%)	
		Training	Test
1	500	74	71
2	1000	81	79
3	1500	86	84
4	2000	93	91

This section shows another example of NAD. The technique proposed in this paper [7] uses genetic algorithms to establish rules for NAD. On this occasion a chromosome in an individual consist of genes matching attributes such as service, flags, super-user attempts and being logged in or not. The tests between training data (Table 1) and test data (Table 2) show promising results as well as a very low false alarm rate. Table 3 shows the comparison between different iterations and their accuracy after evolving the rule sets with GA. These results show that more iterations lead to a significant higher amount of accuracy. The rate of increase in accuracy decreases as the number of iterations increases. Thus, this concludes that there is a reasonable amount of iterations which should be made but further iterations at some point do not increase the accuracy of the result.

4.4 Evaluation

This section evaluates all the discussed techniques presented in this paper. The techniques share the common goal of detecting anomalies however work in a completely different way. Choosing which one to use depends on the resource consumption, robustness, false alarm rates and detection rate. Depending on the situation, the certain technique's adaptability will determine how much information one can learn from training datasets. Analysing the main advantages and disadvantages of each type that was discussed in Chapter 3 and 4, it will allow us to compare the different techniques.

Starting off with statistical-based NAD, these techniques are very susceptible to wrong parameters and hard to adjust. They are reliable when it comes to detection rate and do not rely on training data. However, this means that their only way of adapting is to change the parameters, which may prove to be difficult especially in balancing false positives and negatives. Classification-based techniques are highly adaptive, but rely heavily on training data and do therefore have high detection rates for known anomalies only. Thus, it is challenging to work with classification-based NAD on high-dimensional data. Clustering and outlier based algorithms have the massive drawback since they can only work on continuous data, which restricts them from using data like the protocol or flags mentioned in Chapter 3 (Figure 1-4). Other than that, they can perform well with small-scaled datasets by accurately detecting anomalies. Soft computing methods have the advantage of high adaptability and flexibility. The only disadvantages for these techniques are the reliance on training datasets and the high resource consumption. Knowledge-based techniques profit from high flexibility, scalability and robustness. However, they are reliant on training datasets, which leads to a high detection rate for known anomalies only. It is also challenging to detect rare or unknown anomalies as well as the acquisition of high-quality training datasets. Combination learners are also very dependent on training data, have a high resource consumption and are difficult to adjust. On the other hand, depending on the type of technique used, it can detect both known and unknown anomalies.

5. CONCLUSION

We conclude that Network Anomaly Detection covers a wide variety of interesting features, which algorithms can work with. The algorithms itself show a huge diversity between them, ultimately ending up with one goal: anomaly detection on network data. Altogether it can be said that each method has its own advantages and drawbacks. Therefore, it is challenging to determine which is the best type of algorithm to work. The main reason is that it depends a lot on the user's goal and purpose, availability of the data as well as the amount of available resources.

6. REFERENCES

[1] A. O. Adetunmbi, S. O. Falaki, O. S. Adewale, and B. K. Alese. Network intrusion detection based on rough set and k-nearest neighbour. *International Journal of Computing and ICT Research*, 2(1):60–66, 2008.

[2] J. P. Anderson et al. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, 1980.

[3] D. K. Bhattacharyya and J. K. Kalita. *Network anomaly detection: A machine learning perspective.* CRC Press, 2013.

[4] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: Methods, Systems and Tools. *IEEE communications Surveys & Tutorials*, 16(1):303–336, 2014.

[5] Dennis Schirrmacher, heise.de. Available online at https://www.heise.de/security/meldung/Rekord-DDoS-Attacke-mit-1-1-Terabit-pro-Sekunde-gesichtet-3336494.html ; last accessed on 2017/07/02.

[6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1):18–28, 2009.

[7] M. S. A. Khan. Rule based network intrusion detection using genetic algorithm. *International Journal of Computer Applications*, 18(8):26–29, 2011.

[8] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 25–36. SIAM, 2003.

[9] W. Lee, S. J. Stolfo, and K. W. Mok. A data mining framework for building intrusion detection models. In *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*, pages 120–132. IEEE, 1999.

[10] W. Lu and H. Tong. Detecting network anomalies using cusum and em clustering. In *ISICA*, pages 297–308. Springer, 2009.

[11] M. Mahoney and P. Chan. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In *Recent advances in intrusion detection*, pages 220–237. Springer, 2003.

[12] T. Shon, Y. Kim, C. Lee, and J. Moon. A machine learning framework for network anomaly detection using svm and ga. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 176–183. IEEE, 2005.

[13] Snort. Available online at `http://snort.org`; last accessed on 2017/07/03.

[14] M. Thottan and C. Ji. Anomaly detection in ip networks. *IEEE Transactions on signal processing*, 51(8):2191–2204, 2003.