An Intelligent Defense and Filtration Platform for Network Traffic

Mehrnoosh Monshizadeh[1,2], Vikramajeet Khatri[1][0000-0002-3386-8952], Buse Atli[2] and Raimo Kantola[2]

[1] Nokia Bell Labs, Finland
{mehrnoosh.monshizadeh,vikramajeet.khatri}@nokia-bell-labs.com
[2] Department of Comnet, Aalto University, Espoo, Finland
{mehrnoosh.monshizadeh,buse.atli,raimo.kantola}@aalto.fi

Abstract. Hybrid Anomaly Detection Model (HADM) is a security platform to detect and prevent cyber-attacks on communication networks. The platform uses a combination of linear and learning algorithms combined with protocol analyzer. The linear algorithms filter and extract distinctive attributes and features of the cyber-attacks while the learning algorithms use these attributes and features to identify new types of cyber-attacks. The protocol analyzer in this platform classifies and filters vulnerable protocols to avoid unnecessary computation load. The use of linear algorithms in conjunction with learning algorithms allows the HADM to achieve improved efficiency in terms of accuracy and computation time in order to detect cyber-attacks over existing solutions.

Keywords: Security, Cloud computing, Internet of Things, Machine Learning, Anomaly Detection.

1 Introduction

Although the Intrusion Detection System (IDS) is considered as a well-known mechanism to monitor and detect malicious traffic in communication networks, the cost and high processing time is a challenge to handle large amount of data. Moreover, IDSs and all other detection systems or mechanisms are applicable for known attacks rather than unknown or new attacks.

Data Mining (DM) is a technology that uses highly developed and complex algorithms for processing large volume of data [1]. However, complexity of mentioned algorithms in [1] results to high computation time. In order to solve this problem, network traffic flow control in combination with DM techniques are proposed in this paper. The protocol analyzer in this platform classifies and filters vulnerable protocols to avoid unnecessary computation load. On the other hand, each data set includes hundreds of features that may cause performance degradation in detection process. To overcome this problem, feature selection methods are used to select less number of features and reduce the dimensions of the dataset [1]. In addition, the use of linear algorithms in conjunction with learning algorithms improves accuracy and computation time. Linear algorithms will detect the attack in general level regardless of their types while learning

algorithms cluster the attacks in different categories. This mechanism decreases the load of input data for the learning algorithm that is the most time consuming part because of its complex structure [2].

The rest of the paper is organized as follows. Section 2 briefly reviews related work and research motivation. Section 3 describes the Hybrid Anomaly Detection Model (HADM) architecture. Section 4 discusses the implementation and results. The discussion on future work presented in section 5 and last section concludes the paper.

2 Related Work and Research Motivation

Di Pietro et al. [3] proposes anomaly detection using Deep Packet Inspection (DPI) and machine learning methods upon selected captured packets. Based on output from machine learning method, the packet capture criterion is adjusted again to capture further packets and again fed into DPI and machine learning methods. The machine learning model may use k-Nearest Neighbor (k-NN), replicator nearest neighbor, Bayesian networks, k-means, Artificial Neural Networks (ANN) and Support Vector Machines (SVM) algorithms. However, the learning process of algorithm is not explained and packet capture criterion does not consider taking all packets for one or more protocols.

Vasseur et al. [4] proposes an approach for training supervised learning classifiers for effective detection and cites Deep Neural Networks (DNN) classifier as an example. The study mentions Distributed Denial of Service (DDoS) attacks. Security device in this study refers to firewall, IDS, Intrusion Prevention System (IPS) etc.; which use traffic signatures to access traffic in the network. Distributed learning agents refer to components or modules which use machine learning based anomaly detection to analyze or access traffic in the network. A supervisory device in the network receives traffic from both security device and distributed learning agent; and combines the received traffic to train the classifier. Then, the supervisory device assigns the trained classifier to one or more distributed learning agents. Since, the challenge in the supervised learning classifiers is the labelled training data, this study aims to optimize training by continuously providing the labelled data. However, this study does not consider other machine learning methods, does not use protocol analyzer for filtering the traffic to be analyzed considering the traffic bandwidth and traffic collection period in order to reduce the load.

In the study by Piettro et al. [5], a network trains and generates an expected traffic model based on a set of training data. The device receives an unexpected behavior notification from a particular node based on a comparison between the expected traffic model and an observed traffic behavior by the node. The particular node also prevents the machine learning attack detector from analyzing the observed traffic behavior. The device updates the machine learning attack detector to account for the observed traffic behavior. This method receives a model of normal traffic behavior. The Signature Generator Entity (SGE) compares the input data with the expected normal traffic models. If they are different, SGE generates a signature for the attack class and trains the ANN accordingly with new information for the next detection phase. This model may have high false positive rate since it compares input traffic only with an expected model.

In the study by Yadav et al. [6], network traffic data is collected from several sensors, distributed and installed into network components. The network traffic data comprises process, user and host information. The analytics module inside a network device or Virtual Machine (VM) identifies anomalies within the network traffic data based on dynamic modeling of network behavior. The anomaly detection in this study is based on the honeypot to collect labelled malicious network traffic as an input to the anomaly detection unit comprising of unsupervised and supervised machine learning algorithms. The honeypot relies only on received attacks and not the other attacks. This study points in general dynamic machine learning.

In this study we propose a platform compromises of two main parts where each part independently increases the efficiency of attack detection based on the factors such as precision, recall, accuracy and computation time. While part 1 of the model utilizes some algorithms and the protocol analyzer for traffic filtration, reducing the processing time and increasing the accuracy, the part 2 applies a dynamic feature selection with genetic algorithm to classify unknown attacks and increase the accuracy as well.

The HADM model comprises a protocol analyzer, linear and learning algorithms as well as other modules. Since some protocols such as streaming protocols are not vulnerable and attackers usually target specific protocols, protocol analyzer in this platform classifies and filters vulnerable protocols to avoid unnecessary computation load. Protocol analyzer forwards the filtered traffic either to a linear algorithm only for Denial of Service (DoS) detection or to a combination of a linear and a learning algorithm for other types of attacks. The linear algorithm initially defines if the traffic is secure or unsecure regardless of the attack type. In addition, it extracts the proper features in order to provide them to a learning algorithm in order to classify already known attacks and detect unknown attacks. The other counter measurement located after the learning algorithm extracts information for known attacks (which network is already protected against them) from other deployed security mechanisms in the network e.g., firewall, IDS, DPI etc. It compares the extracted information with the attack received from the linear algorithm and drops the similar attacks. In each step, a feedback is sent to a database for next level detection. In the learning algorithm, the received attack is assigned to one of the attack clusters. In addition, algorithm changes its structure and input weights dynamically based on the received feedback. If the attack does not belong to any of the mentioned clusters, it is assigned to a totally new cluster. This novel mechanism dynamically defines new features in order to detect new types of attack.

3 Architecture

As it is shown in Fig. 1, the HADM functionality is divided into three phases: protocol analyzer, dynamic machine learning and validator & database.

3.1 Protocol Analyzer

The first phase of the proposed HADM is called protocol analyzer which filters vulnerable protocols. The protocol refers to communication protocol over which the traffic is

carried on such as HyperText Transfer Protocol (HTTP) and Transmission Control Protocol (TCP). As it is shown in Fig. 2, protocol analyzer consists of five modules.

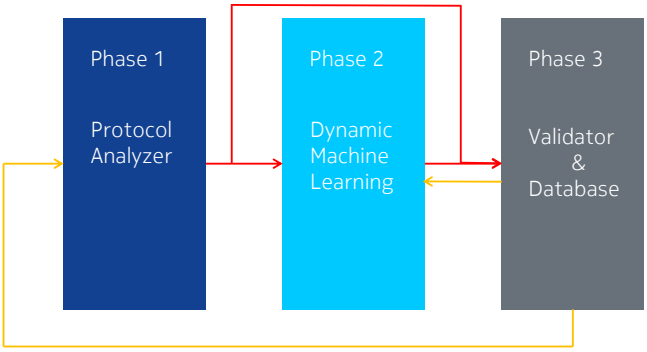

Fig. 1. HADM on operational level in brief

Decision module. This module includes a list of vulnerable protocols which are predefined and also dynamically updated based on the received feedback from log file via database. Some protocols such as HTTP and TCP are well known vulnerable protocols while others like Real Time Streaming Protocol (RTSP) could be a safe protocol. It checks whether traffic is carried on any of the listed vulnerable protocol.

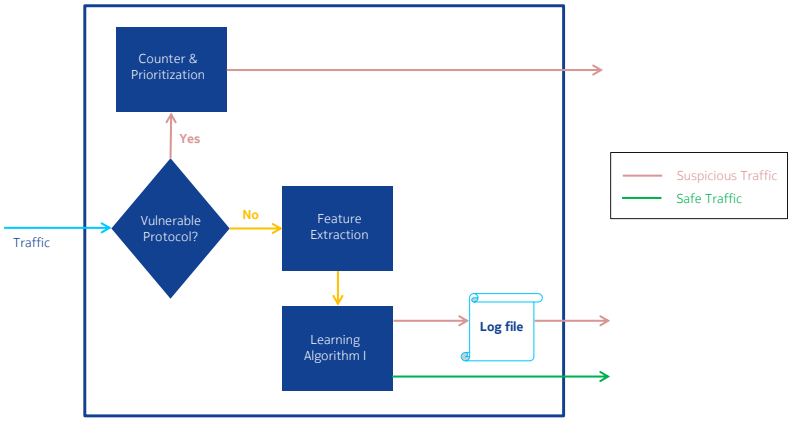

Fig. 2. Protocol Analyzer

Counter and Prioritization module. The function of this module is based on the occurrence threshold (n) and prioritization. It means that if the vulnerable protocol carries suspected traffic for n times, then this module will forward the suspicious traffic to the next layer for detection and labelling. The idea is to cycle all possible vulnerable protocols over an agreed time window (1 hour, 1 day etc.). The module only keeps a certain number of vulnerable protocol in the list which is based on prioritization. For example,

if we already have 20 vulnerable protocols and 21st comes up, then the counter must prioritize only 20 of these protocols. The prioritization is based on the counting occurrence over the time window. The sole purpose of this technique is to reduce the computation load of traffic analysis.

Feature Extraction. It extracts the best features from the suspicious protocol. This module is utilized in second phase as well.

Learning Algorithm I. If the protocol (that carries the input traffic) is not listed as vulnerable, traffic is still sent to this learning algorithm for analysis and reconfirmation. The learning algorithm I will check whether the protocol is vulnerable or not. Our proposed platform is tested with Extreme Learning Machines (ELM), Self-Organizing Map (SOM) and MultiLayer Perceptron (MLP) algorithms.

Log file. Every time the learning algorithm I in the protocol analyzer detects a new vulnerable protocol, it is recorded into the log file and a feedback will be sent to decision module via database. The log file records packet features such as time stamp, packet size, Internet Protocol (IP) header and information on other layers (Ethernet, TCP, application layer).

3.2 Dynamic Machine Learning

The second phase of the proposed HADM combines linear and learning algorithms for efficient attack detection. As it is illustrated in Fig. 3, dynamic machine learning consists of the following modules in addition to feature extraction that has been explained earlier.

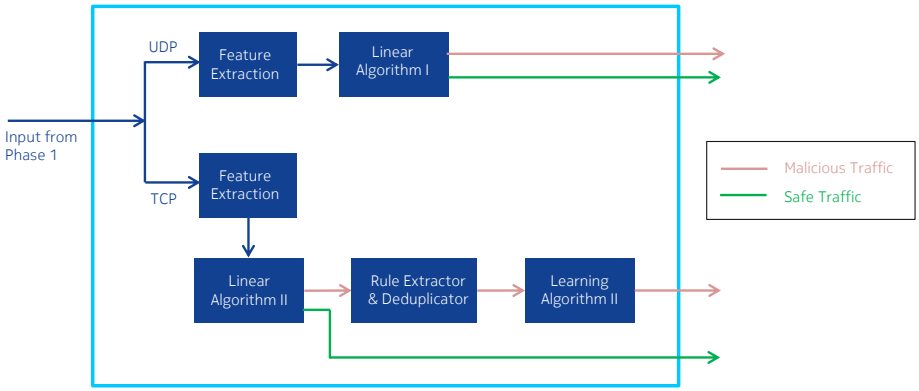

Fig. 3. Dynamic Machine Learning

Linear Algorithm I. This module analyzes User Datagram Protocol (UDP) traffic to detect UDP DoS attacks. Therefore, a separate algorithm such as Decision Tree (DT)

is considered for this module in order to avoid overloading the rest of the proposed hybrid model. However, the decision tree algorithm can be replaced based on the operator demand, we have considered it due to its low processing time.

Rule Extractor and Deduplicator. This module filters the known attacks that system is already protected against, by using other deployed security mechanisms and forwards other attacks to learning algorithm II for labelling. A set of rules are extracted from those deployed security mechanisms in the network, the extracted information is compared with received attack from linear algorithm I and is dropped if it is similar. Rules in this module are updated dynamically based on input from the parallel security mechanisms.

Learning Algorithm II. This module is the last detection layer. Initial features and clusters are defined for the algorithm during the training process in order to cluster different attacks such as Botnet attack (B) and Malicious codes (M). At first, the traffic is labelled to one of the clusters based on their similarity or distance. Since the traffic that arrives to this module has been already identified as attack, if it does not belong to any of the mentioned clusters then it is considered as new type of attack (N) and a cluster will be created for it. The features of the new type of attack (N) must be added to the algorithm accordingly. The implementation of the proposed platform with Artificial Neural Networks (ANN) and Genetic Algorithm (GA) is already ongoing by authors and results will be presented in future paper. Other potential unsupervised algorithms can be SOM and hierarchical clustering.

3.3 Validator & Database

The last phase of the proposed HADM validates detected attacks, stores them into the database and shares the updates to all relevant modules. It consists of following modules as shown in Fig. 4.

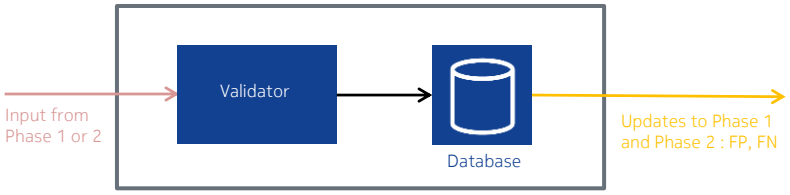

Fig. 4. Validator & Database

The validator acts similar to error detection module in order to decrease False Positive (FP) and False Negative (FN) rates. If the actual result differs from expected result then result is considered as error and is not registered in the database (DB). Validator should be always updated with labeled data and output from detection algorithms. The database saves all the results of detection algorithms; from each algorithm, a sample of

the outcome (feedback) is sent to database that will be used in the future detection. A database contains known attacks, new attacks and dropped attacks.

4 Implementation Phases

As it is shown in Fig. 5, the implementation of HADM platform is divided into two parts. Due to space limitation, the part 1 is presented in this paper while the implementation and experimental results of part 2 will be published in a separate paper. In this section, we first introduce the algorithms and feature extraction methods that are applied for mentioned algorithms. Next, we describe the selected datasets for the experimental study. Finally, we present the measures employed to evaluate the performance of each algorithm independently and also for the integrated scenario.

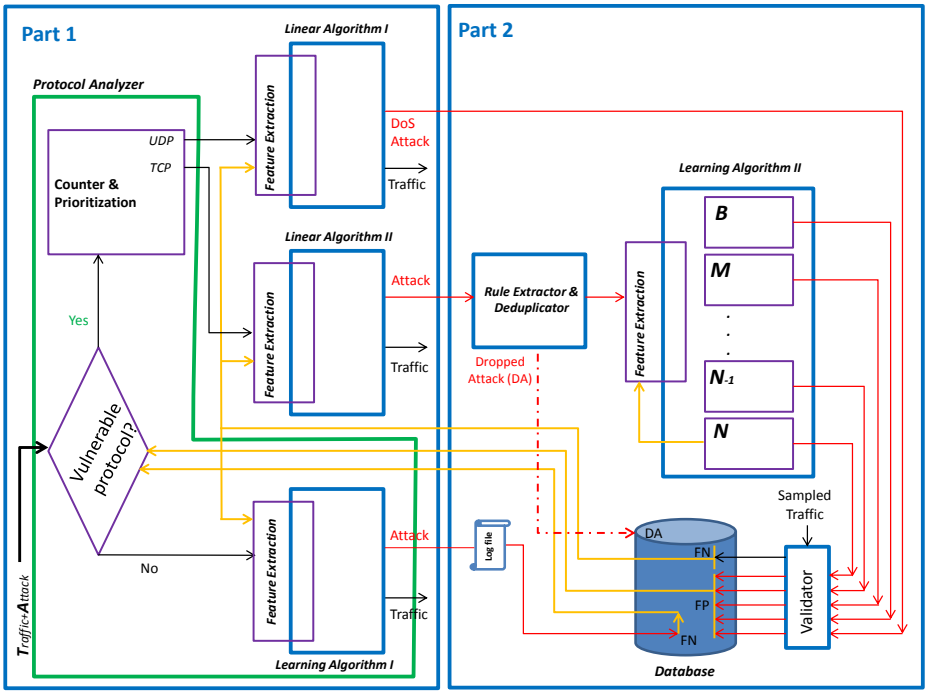

Fig. 5. Hybrid Anomaly Detection Model (HADM)

4.1 Applied Algorithms

In order to evaluate HADM, different algorithms including ELM, MLP (with 10 and 50 hidden nodes), k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), Decision Tree (DT) and Logistic Regression (LR) are applied. In addition, our simple Decision maker algorithm (Algorithm D) is used in protocol analyzer to filter vulnerable protocol. The algorithm D is shown in Fig. 6.

```
for each packet do:
   if packet is encapsulated then
      decapsulate packet
   else
      if packet contains vulnerable protocol then
         if packet is carried over UDP then
            forward packet to Linear Algorithm I
         elseif packet is carried over TCP then
            forward packet to Linear Algorithm II
      end if
      elseif packet does not contain vulnerable protocol then
         forward packet to Learning Algorithm I
      end if
   end if
end for
```

Fig. 6. Pseudocode for algorithm D

4.2 Feature selection

The original dataset contains 27 features as shown in Table 1.

Table 1. Features in datasets

No.	Feature	No.	Feature	No.	Feature
1	Ethernet size	10	IP destination	19	Connection starting time
2	Ethernet destination	11	TCP source port	20	IP fragmentation flag
3	Ethernet source	12	TCP destination port	21	IP fragmentation overlap
4	IP header length	13	UDP source port	22	TCP ACK flag
5	IP type of service	14	UDP destination port	23	TCP retransmission
6	IP length	15	UDP length	24	TCP push flag
7	IP time to live	16	ICMP type	25	TCP SYN flag
8	IP protocol	17	ICMP code	26	TCP FIN flag
9	IP source	18	Duration of connection	27	TCP urgent flag

Since there are not many variations on features related to protocol, for learning algorithm I in protocol analyzer module, we considered 9 fixed features including source IP address (saddr1, saddr2, saddr3, saddr4), destination IP address (daddr1, daddr2, daddr3, daddr4) and time to live (ip.ttl). Three different feature selection methods including Chi2, FScore and SVMonline have been applied on linear algorithm I and II and the best combinations for both feature selection methods and algorithms have been selected based on the achieved efficiency. The final combinations include FScore for

LR and Chi2 for k-NN. The selected best features for these algorithms can be seen in Table 2.

Table 2. Selected features for each algorithm

Linear Algorithm I (k-NN)				Linear Algorithm II (LR)			
No.	Feature	No.	Feature	No.	Feature	No.	Feature
1	SDC2	6	dstport	1	daddr4	6	saddr2
2	daddr2	7	srcport	2	SMC3	7	saddr4
3	SMC1	8	daddr4	3	ip.len	8	SDC1
4	saddr2	9	saddr4	4	frame.len	9	daddr2
5	SMC3	10	ip.ttl	5	daddr3	10	SMC2

4.3 Data Preprocessing

The ISCX-2012 dataset exhibits realistic network behavior and contains diverse intrusion scenarios. The dataset includes network traffic on HTTP, SMTP, SSH, IMAP, POP3, and FTP protocols with FTP and SSH password brute force, Java based Meterpreter, Add new Superuser, Linux Meterpreter payload and C100Webshel attacks [7]. To evaluate HADM efficiency, the modified version of the ISCX-2012 [8] dataset with diverse attacks have been used. Since the ISCX-2012 dataset lacks the DoS attack on UDP protocol, we extracted the UDP DoS from UNSW-NB15 dataset [9] and injected to the ISCX-2012 dataset. The final dataset is classified in three categories: normal, attack and unknown. The IP address and hex MAC address of the applied dataset are transformed into separate numeric attributes to improve the performance of the algorithms.

Table 3 shows the distribution of each intrusion type in the training and testing data.

Table 3. Distribution of packets in dataset

Data	Normal	DoS	Other attacks	Unknown	Total
Training (2/3)	1419441	47198	131713	2359518	3957870
Testing (1/3)	709723	23599	65856	1179763	1978941

4.4 Experimental Results

All the experiments are carried out on a workstation with Intel core i5 Quad @2.6GHz, 16 GB RAM, 500GB HDD. The scripts were developed in Python in a Linux environment. All applied algorithms in the evaluation process of HADM are trained once and saved for the future tests. However, currently platform is tested on single workstation, the whole functionality can be installed on several VMs for load balancing and decentralized monitoring purposes in order to handle large amount of traffic at core network. The similar mechanism has been explained in other papers presented by authors on SDN security [10].

As it is shown in Table 4, for learning algorithm I, testing time, total accuracy score, binary cross entropy error and false negative score are compared and MLP methods outperforms the ELM approach.

Table 4. Learning algorithm I performance evaluation

Data	Accuracy score	Cross entropy error	False Negative score	Testing time (s)
ELM 10	0.80	0.84	2.35	0.72
ELM 50	0.58	3.01	0.95	1.06
MLP 10	0.87	0.66	0.11	0.09
MLP 50	0.88	0.57	0.12	0.34

The binary cross entropy error and the false negative score of ELM is quite high, which means that in most cases, it fails to give alarm when an intrusion occurs. If MLP with 10 and 50 hidden layer neurons are compared, it can be seen that MLP with 50 hidden layer neurons performs slightly better than the MLP with 10 hidden layer neurons in terms of differentiating the normal and attack traffic, although the time complexity for the testing is smaller with MLP with 10 hidden layer neurons. Since the overall architecture has a high time and model complexity with many preprocessing (protocol analyzer) and post-processing (genetic algorithm etc.) steps, we have chosen MLP with 10 hidden layer neurons for this module to reduce the overall processing time for labelling an incoming network packet and the model complexity.

To evaluate part 1 of HADM as whole and with protocol analyzer functionality, the traffic carried on vulnerable protocol will be directed to k-NN and LR and the rest of the traffic to ELM/MLP. Training in this phase is covered in two scenarios. In the scenario I, the incoming traffic to k-NN includes: UDP_{DoS}+TCP_{DoS} while the LR algorithm processes the rest of the traffic (TCP-TCP_{DoS}). In Scenario II, all UDP traffic will be forwarded to k-NN algorithm and all TCP traffic to LR algorithm. Table 5 and Table 6 show the performance evaluation for testing scenario I and II based on three metrics. The precision and recall values are measured for attack class.

As it is shown in Table 5, Table 6 and Table 7, the computation time decreased greatly in scenario II, while the accuracy, precision and recall factors also improved. That means the HADM outperforms considerably while UDP DoS is separated from attacks carried over TCP (applying protocol analyzer). However, in this study, we only classified our input data to three categories of normal, attack and unknown and have not applied the part II of HADM yet to categorize different types of attacks, which is a factor to consider in our future work.

Table 5. k-NN and LR detection performance for scenario I

Input data	Accuracy	Precision	Recall
k-NN	0.94	0.95	0.93
LR	0.92	0.95	0.06

Table 6. k-NN and LR detection performance for scenario II

Input data	Accuracy	Precision	Recall
k-NN	0.99	0.97	0.97
LR	0.92	0.63	0.20

Table 7. Computation time

Input data	MLP 10		k-NN		LR	
	Training	Testing	Training	Testing	Training	Testing
Scenario I	158.08	0.24	10926.64	5766.36	80.63	0.62
Scenario II	158.08	0.24	284.63	105.23	80.6	0.44

5 Discussion on Future Work

The proposed model compromises of two main parts where each part independently increases the efficiency of attack detection based on the factors such as precision, recall, accuracy and computation time. While part 1 of the model utilizes the protocol analyzer and mentioned algorithms are implemented for traffic filtration, reducing the processing time and increasing the accuracy, the part 2 applies a dynamic feature selection with genetic algorithm in order to classify unknown attacks and increase the accuracy as well. In current paper, in order to evaluate the performance of the part 1, the modified version of ISCX-2012 and UNSW-N15 datasets with diverse attacks have been used. The performance of the part 1 is compared for two scenarios where UDP DoS is separated from other protocols and where it is not.

In order to evaluate the model robustness and scalability, we will also test the model against five different latest datasets to cover also the new protocols that are missed from ISCX-2012 [11]. Furthermore, for each algorithm, we applied other feature selection methods such as FScore, Chi2 and SVM_{online} to find the best features [12]. On the other hand, we applied three more different algorithms including MultiLayer Perceptron (MLP), SVM and Decision Tree (DT). The best algorithms were selected through a benchmark on applied datasets and based on the achieved accuracy, FP, FN, training and testing time.

6 Conclusion

This paper reviewed current mechanisms such as Data Mining (DM) techniques for security purposes and their limitations to defend networks against cyber-attacks. While learning algorithms may have high accuracy in general and longer computation time in comparison to linear algorithms, they may give unexpected responses to a specific type of attack. Therefore, this paper proposed a security platform called Hybrid Anomaly Detection Model (HADM) that uses a combination of linear and learning algorithms

along with protocol analyzer. The linear algorithms filter and extract distinctive attributes and features of the cyber-attacks, while the learning algorithms use these attributes and features to identify new types of cyber-attacks. The protocol analyzer classifies and filters vulnerable protocols to avoid unnecessary computation load. The use of linear algorithms in conjunction with learning algorithms allows the HADM to achieve improved efficiency in terms of the accuracy and computation time to detect cyber-attacks over existing solutions.

We also described testing scenarios and concluded that the proposed protocol analyzer filters the vulnerable protocols such as UDP and TCP to decrease the computation load, processing time and accuracy for applied algorithms. Our future work concentrates on applying extra datasets, feature selection methods and algorithms to evaluate the model robustness and scalability.

References

1. Desale, K., Ade, R.: Genetic algorithm based feature selection approach for effective intrusion detection system. In 2015 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2015, pp. 1-6.
2. Monshizadeh, M., Yan, Z.: Security Related Data Mining. In: IEEE International Conference on Computer and Information Technology, Xi'an, pp. 775-782 (2014).
3. Di Pietro, A. et al.: Dynamic deep packet inspection for anomaly detection. US Patent 2017099310 (A1), 6 Apr. 2017.
4. Vasseur, J. et al.: Anomaly detection in a network coupling state information with machine learning outputs. US Patent 20170104774 (A1), 13 Apr. 2017.
5. Di Pietro, A. et al.: Signature creation for unknown attacks. US Patent 20160028750 (A1), 28 Jan. 2016.
6. Yadav, N. et al.: Network behavior data collection and analytics for anomaly detection. US Patent 20160359695 (A1), 8 Dec. 2016.
7. Atli, B.: Anomaly-Based Intrusion Detection by Modeling Probability Distributions of Flow Characteristics. MS Thesis. Aalto University (2017).
8. ISCX-2012, University of New Brunswick, http://www.unb.ca/cic/datasets/ids.html
9. The UNSW-NB15 data set, https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/
10. Monshizadeh, M., Khatri, V., Kantola, R.: Detection as a service: An SDN application. In: 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, pp. 285-290 (2017).
11. Sharafaldin, I. et al.: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018
12. Yann, L. et al.: Deep learning. In nature 521.7553 (2015): 436