Anomalies Classification Approach for Network-based Intrusion Detection System

Qais Saif Qassim, Abdullah Mohd Zin, and Mohd Juzaiddin Ab Aziz
(Corresponding author: Qais Saif Qassim)

Research Center for Software Technology and Management, Information Science and Technology University
Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia
(Email: qaisjanabi@gmail.com)

(Received Sep. 1, 2015; revised and accepted Dec. 7, 2015 & Jan. 15, 2016)

Abstract

Anomaly based intrusion detection system (A-IDS) is considered to be a better option than signature based system since it does not require prior knowledge of attack signature before it can be used to detect an intrusion. However managing alarms generated by this system is more difficult than signature-based intrusion detection systems (S-IDSs). This is due to the fact that S-IDS generates rich information along with the reported alarms whereas A-IDS may just identify the connection stream that is detected as malicious. A-IDS raises an alarm every time it detect an activity that deviates from the baseline model of the normal behaviour. Therefore, the cause of the anomaly itself is unknown to the intrusion detection system. This brings in a substantial challenge problem in managing IDS alarms and recognizing false positive from true alarms. Therefore, determining the class of an attack detected by anomaly-based detection systems is a significant task. This paper serves two folds; firstly, it presents a set of network traffic features that deemed to be the most relevant features in identifying wide range of network anomalies. Secondly, the paper presents an A-IDS alarm classifier based on machine learning technologies to automatically classify activities detected by a packet header-based anomaly detection system. Evaluation experiments showed that machine learning algorithms are capable of classifying malicious activities in an effective and efficient means.

Keywords: Alarm classification, anomaly-based, feature selection, machine learning

1 Introduction

Anomaly-based detection system is designed to uncover abnormal patterns of behaviors, in which anything that widely deviates from normal usage patterns will be considered as an intrusion [4]. It is considered to be a better option than signature based system since it does not require prior knowledge of attack signature before it can be used to detect an intrusion. However, identifying the class of attack poses a significant problem in anomaly based IDS. In signature based IDS, this process is trivial since each signature is a result of an analysis of the corresponding attack conducted by security experts; in which the attack class is manually assigned during the signature development process [9, 12]. Unlike signature-based IDS, the anomaly-based detection system cannot associate the detected activity with an attack class. In fact one of the major weaknesses of anomaly-based intrusion detection system is that, it cannot classify the detected activity to determine the severity level and the consequences of the detected activity [10].

By classifying an attack, it is possible to set default actions for handling a certain alarm. As well as, in order to estimate the risk of unknown attacks, a solution to automate the classification of anomaly-based alarms is required. However, so far no effective and efficient automatic or semi-automatic approach that is currently available, able to classify anomaly-based alarms at runtime [15, 31]. Thus, any anomaly-based alarm must be manually processed to identify its class; this may increase the workload of security analyst, and will effectively increase time required; as well as, the dependence on security analysts. Another limitation of manual alarm processing is that the complexity and dynamically changing traffic statistics may introduce the possibly of human error. This paper presents Network Anomalies Classifier (NAC) that uses machine learning technologies to automatically classify activities detected by a packet header-based anomaly detection system.

The rest of this paper is organized as follows: Section 2 presents an overview of the current state of attack identification and classification addressing the feature sets have been monitored, Section 3 presents the attack scenarios providing the common network traffic features to be monitored to identify different attack classes, Section 4 describes the research methodology, Section 5 discusses the evaluation of the proposed system and Section 6 presents the conclusions and future works.

2 Related Works

IDS alarm classification has been an active research area for the past few years, recent researchers have focused on managing the generated alarms to identify real threats from false alarms and to classify the alarms into distinct classes. Several methods have been proposed to analyse the reported alarms based on different classification algorithms and network traffic features [1]. This section presents some of the recently proposed methods.

Entropy based analysis [21] have been employed to analyze a signature-based IDS alarms (more specifically, Snort) and detect real network attacks. The proposed method uses Shannon entropy to examine the distributions of five statistical features of Snort alarms as illustrated in Table 1. The features used are; the number of alarms generated from each distinct source IP address, the number of alarms sent to a destination host, source and destination threats' severity grade and datagram length. An adaptive false alarm filter [23] have been utilized to filter out false alarms with the best machine learning algorithm based on distinct network features. The Authors have intended to reduce the false alarms generated by signature-based IDS (Snort) in real time, and have selected 8 network features to represent the generated alarms as follows; Snort's description of an attack, attack classification, priority of an attack, packet type, source IP address, source port number, destination IP address and destination port number. They have used DARPA dataset to evaluate six different machine learning algorithms; K-nearest neighbor, support vector machines, naive bayes, neural networks and decision trees using Weka platform. And then, they have designed an adaptive false alarm filter to select the best single-performance algorithm in filtering out false alarms.

An approach of semi-supervised learning mechanism have been introduced by Chiu [3] to build an alarm filter for signature-based intrusion detection system. The authors have selected eight network features specifically; the connection's start time, the connection's duration, local and remote IP addresses that participated in the connection, connection's service, local and remote ports used by the connection, the number of bytes sent and received and the state of the connection. In [27] the author has used Lincoln laboratory dataset to find suitable subsets of features for network attack detection. The feature subsets were formed using prior knowledge from previous IDS researches and in addition, from analysing network attacks and their effect to the traffic flows, the selected features are illustrated in Table 1. The author showed that attacks of similar type, have similar effect to the network traffic and thus, subsets of features were formed for each attack type.

Flow-based analysis has been considered by Knuuti [16]. The author has compared the usability and performance of three different intrusion detection systems based on the identified network traffic flow features. The evaluated systems were Snort, Bro-IDS and TRCNetAD. Snort and Bro-IDS are signature-based intrusion detection systems while the later is an anomaly-based IDS. The features set that the author used are as illustrated in Table 1, which are statistical representations of the network traffic flow. The study conducted two, one week long, traffic capturing periods to collect data for the evaluation. Using the selected features, Snort was able to detect over 1.5 million intrusions during the one-week traffic capturing period. Snort was able to detect buffer overflow attacks, Trojan, denial of service, VoIP attacks, Heap overflow attacks, DNS spoofing attack and spyware. Bro-IDS detected approximately eight thousand intrusions which were address and port scan. TRCNetAD detected 150 thousand anomalies during the same time period.

Rule adaptation approach in managing IDS alarms have been considered by Lin [20]. The study has proposed a Weighted Score-based Rule Adaptation (WSRA) mechanism; which have the facility to learn from expert's feedback. Features used in this work are illustrated in Table 1 and as follows; total number of source and destination IP addresses in defined time window, source and destination port number, snort's signature, attack class, and timestamp.

Monitor deviations in network traffic features distributions from baseline model had been considered in IDS alarm management approaches [5]. The study analysed events that affect the distribution of traffic features and mark them as anomalies. The proposed system monitored network-wide backbone traffic using the features listed in Table 1. They have monitored the changes on the four IP packet header features between traffic flows using different algorithms. However, the study didn't evaluate the proposed method in real network traffic.

3 Feature Selection Based on Attack Scenarios

Feature selection is an important step in building intrusion detection and constructing alarm classification modules. During feature selection phase, a set of network traffic attributes or features deemed to be the most effective attributes is extracted in order to construct suitable classification module [29, 33]. A key challenging problem that many researchers face is how to choose the optimal set of features [1, 28], as not all features are relevant and have an impact on the classification performance, and in many cases, irrelevant features can impact the classification accuracy and cause slow training and testing processes. By analysing known attacks and their influence to the normal network traffic, it is possible to define which traffic features are relevant and therefore should be monitored. The idea behind this approach is to define the characteristics of a specific attack category. This is done by analysing

Table 1: Network traffic features used in prior studies

Study	Features Used	Num. of Features
[21]	The number of alarms generated from each distinct source IP address, the number of alarms sent to a destination host, source and destination threats' severity grade and datagram length	5
[23]	Description of the attack, Snort's classification, Alarm priority, packet type, source IP address, source port number, destination IP address and destination port number.	8
[3]	The connection's start time, the connection's duration, local and remote IP addresses that participated in the connection, connection's service, local and remote ports used by the connection, the number of bytes sent and received and the state of the connection	8
[27]	IP address, timestamp, number of receiving sequences, number of receiving sequences from different IP's, number of sending sequences, number of sending sequences to different IP's, amount of data received, amount of data sent, amount of packets received, amount of packets sent, number of different port numbers used over 1024, number of port numbers used over 1024, number of different port numbers used below or at 1024, number of port numbers used below or at 1024, number of UDP flows, number of TCP connections, number of ICMP packets, number of SMTP connections, number of FTP connections, number of HTTP connections, number of DNS connections, number of Telnet connections, number of SSH connections	24
[20]	Total number of source and destination IP addresses in defined time window, source port number, destination port number, snort's signature, attack class, and timestamp.	5
[16]	IP address, timestamp, number of ICMP packets, number of UDP flows, number of TCP connections, amount of received data, amount of sent data, number of received packets, number of sent packets, number of different port numbers used over 1024, number of port numbers used over 1024, number of different port numbers used below 1024, number of port numbers used below 1024, number of receiving sequences from different IP's, number of receiving sequences, number of sending sequences to different IP's and number of sending sequences.	17
[5]	Source IP address, destination IP address, source port number and destination port number.	4

the attacks classification done by MITRE Corp [24]. Researchers at MITRE Corp. have developed attack taxonomy for the United State Department of Homeland Security [7]; the main goal of this taxonomy is to create a list of patterns employed by attackers when compromising information systems, along with a comprehensive schema and classification taxonomy [34]. The project entitled as the Common Attack Pattern Enumeration and Classification (CAPEC). The classification in CAPEC is based on the mechanism used to attack that include; resource depletion, network reconnaissance, spoofing, exploitation of authentication, and exploitation of privileges.

3.1 Resource Depletion (DOS)

An attacker depletes a resource to the point that the target's functionality is affected. The result of a successful resource depletion attack is usually the denial of one or more services offered by the target [11, 19]. In order to deplete the target's resources the attacker must interact with the target and a client or script capable of making repeated requests over a network. If the attacker has some privileges on the system the required resource will likely be the ability to run a binary or upload a compiled exploit, or write and execute a script or program that consumes resources. Most of resource depletion attacks are detectable by monitoring from the traffic flows and the amount of data sent by the source. Therefore, the features that should be monitored for resource depletion attacks are as follows [26, 27];

1) Number of sequences received during the observation period;
2) Amount of bytes received during the observation period;
3) Total number of packet received;
4) Total number of sequences received during the observation period from different IP's;
5) Number of sequences sent during the observation period;
6) Amount of bytes sent during the observation period;
7) Total number of packet sent;
8) Total number of sequences sent during the observation period to different IP's;
9) Total number of different TCP and UDP port numbers used by source;
10) Total number of different TCP and UDP port numbers used by the host;
11) Number of TCP requests for transmission;
12) Number of half open connections;
13) Number of established connections which represents an open connection;
14) Number of connection termination requests sent;
15) Number of confirming connection termination received;
16) Total number of TCP connections during the observation period;
17) Total number of UDP flows during the observation period;
18) Total number of TCP connections initiated by source;
19) Total number of UDP flows received;
20) Total number of TCP connections initiated by the host;
21) Total number of UDP flows sent.

3.2 Network Reconnaissance (Probe)

An attacker engages in network reconnaissance operations to gather information about a target network or its hosts. Network Reconnaissance techniques can range from stealthy to noisy and utilize different tools and methods depending upon the scope of the reconnaissance [24, 26]. Host discovery and port scanning are common examples of network reconnaissance, where the attacker tries to map out IP addresses and operating systems that are in use, as well as what services the hosts are providing [14]. In general, in network reconnaissance operations the attacker tries to find out all the possible means and methods that it can use to perform other attacks such as denial of service or gaining an unauthorised access to the inner network. Most of network reconnaissance attacks are detectable by monitoring from the traffic flows. Therefore, the features that should be monitored for such attacks are as follows [17, 23];

1) Number of sequences received during the observation period;
2) Total number of sequences received during the observation period from different IP's;
3) Number of sequences sent during the observation period;
4) Total number of sequences sent during the observation period to different IP's;
5) Total number of different TCP and UDP port numbers used by source;
6) Total number of different TCP and UDP port numbers used by the host;
7) Number of half open connections;
8) Number of connection termination requests sent;
9) Number of confirming connection termination received;
10) Total number of TCP connections during the observation period;
11) Total number of UDP flows during the observation period;
12) Total number of TCP connections initiated by source;
13) Total number of UDP flows received;
14) Total number of TCP connections initiated by the host;
15) Total number of UDP flows sent.

3.3 Spoofing

An attacker interacts with the target in such a way as to convince the target that it is interacting with some other principal and as such take actions based on the level of trust that exists between the target and the other principal [30]. Many of the protocols in the TCP/IP suite do not provide mechanisms for authenticating the source or destination of a message. They are thus vulnerable to spoofing attacks when extra precautions are not taken by applications to verify the identity of the sending or receiving host. IP spoofing may be used to leverage man-in-the-middle attacks against hosts on a computer network. Spoofing attacks which take advantage of TCP/IP suite protocols may be mitigated with deep packet inspection. The features that should be monitored for such attacks are as follows [25];

1) Total number of sequences received during the observation period from different IP's;
2) Total number of sequences sent during the observation period to different IP's;
3) Number of privileged port numbers used during the observation period;
4) Number of different privileged port numbers used during the observation period;
5) Number of registered ports used during the observation period;
6) Number of different registered port numbers used;
7) Total number of different TCP and UDP port numbers used by source;
8) Total number of different TCP and UDP port numbers used by the host;
9) Number of TCP requests for transmission;
10) Number of half open connections;
11) Number of established connections which represents an open connection;
12) Number of connection termination requests sent;
13) Number of confirming connection termination received.

3.4 Exploitation of Authentication

An attacker actively targets exploitation of weaknesses, limitations and assumptions in the mechanisms a target utilizes to manage identity and authentication. Such exploitation can lead to the complete subversion of any trust the target system may have in the identity of any entity with which it interacts. The exploitation of authentication attacks are detectable from the payload data by looking for specific patterns. Some of the attacks are though also detectable from the network traffic by looking for malformed packets that are oversized, fragmented or using, for example, abnormal TCP flag options [22]. Therefore, the features that should be monitored for such attacks are as follows;

1) Total number of sequences received during the observation period from different IP's;
2) Number of privileged port numbers used during the observation period;
3) Number of different privileged port numbers used during the observation period;
4) Number of registered ports used during the observation period;
5) Number of different registered port numbers used;
6) Number of half open connections;
7) Total number of TCP connections during the observation period;
8) Total number of UDP flows during the observation period;
9) Total number of TCP connections initiated by source;
10) Total number of UDP flows received;
11) Total number of TCP connections initiated by the host;
12) Total number of UDP flows sent.

3.5 Exploitation of Privilege/Trust

An attacker actively targets exploitation of weaknesses, limitations and assumptions in the mechanisms a target utilizes to manage access to its resources or authorize utilization of its functionality. Such exploitation can lead to the complete subversion of any control the target has over its data or functionality enabling almost any desired action on the part of the attacker. Similarly to exploitation of authentication attacks, this type of attacks detectable from the payload data by looking for specific patterns. However, some of the attacks are though also detectable from the network traffic. Therefore, the features that should be monitored for such attacks are as follows [35, 36];

1) Total number of sequences received during the observation period from different IP's;
2) Total number of sequences sent during the observation period to different IP's;
3) Number of privileged port numbers used during the observation period;

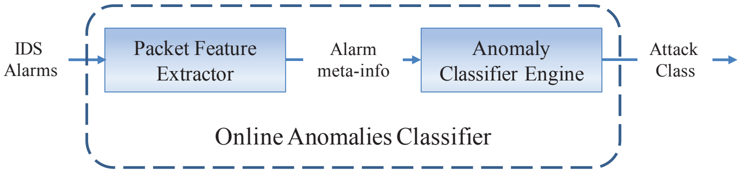

Figure 1: Online anomalies classifier

4) Number of different privileged port numbers used during the observation period;

5) Number of registered ports used during the observation period;

6) Number of different registered port numbers used;

7) Number of half open connections;

8) Total number of TCP connections during the observation period;

9) Total number of UDP flows during the observation period.

4 Network Anomalies Classifier (NAC)

This section presents an A-IDS alarm classification method which relies on machine learning algorithm and attack examples learnt from S-IDS during the training process. The proposed method monitors the network communication pattern and actively extracts the required network traffic features. The proposed system analyse IDS alarms and attempt to classify them based on prelearnt classification model. The classification model is constructed based on attack examples supplied during training phase, during the training phase Snort have been used to provide alarm class definitions of the activities detected by the anomaly detection system. The proposed system is represented by the Network Anomalies Classifier (NAC) module depicted in Figure 1. The NAC is responsible for an automatic classification of activities detected by a packet header-based anomaly detection system (specifically, PHAD) based on predefined set of patterns of attack mechanisms.

The proposed network anomalies classifier uses machine learning algorithm to assign class labels to the detected activities. The NAC consist of two interacting components; the Packet Features Extractor (PFE) and Anomaly Classifier Engine (ACE) as illustrated in Figure 1. The PFE monitors network traffic flow and extracts traffic flow features to generate alarm meta-information as a vector representing symptoms vector. The symptoms vector then, to be directed to the anomaly classifier engine for further analysis. The most suitable traffic flow features are selected by handpicking from the feature spectrum based on the prior knowledge about the environment that the IDS is monitoring and the analysis of known attack types.

The ACE is responsible for automatically classify the detected activity and determine the attack class. Before the classifier engine is able to classify new incoming alarms automatically, the ACE is trained with several types of attack meta-information to build a classification model. During the training phase, the attack meta-information is provided automatically by extracting specific information from known attack signatures. In this work, a signature-based IDS is deployed next to the anomaly detection system and both monitor the same network traffic. Consequently, the S-IDS is responsible to feed the NAC with the attack class of any alarm generated by the two systems.

4.1 Packet Features Extractor (PFE)

Network traffic contains features that are redundant or their contribution to the classification process is little. Therefore, it is essential to choose among the data what is relevant to consider and what is not [8]. By reducing the amount of features, the classifier's computational speed is improved and the overall performance is increased. Thus, Feature selection plays an important role when creating a model of the network traffic. The features should represent the traffic data as accurate as possible. The challenge is on discovering the most suitable features having major contribution to the classification process [4].

Network traffic is collected based on either packet data or network traffic flow, each provides a different type of visibility and collectively can provide a complete view of the network activities. As data streams flow across the network, the network packet-based sniffer captures each packet and decodes the packet's raw data, showing the values of various fields in the packet. The network traffic contains users' confidential information [33]. Consequently, a deep packet analysis cannot be done, and only limited analysis for the network traffic can be achieved. Therefore, the header fields of the packets can be checked, but not the user's data in the payload.

A traffic flow can be described as; all network packets with the same source/destination IP address, source/destination ports, protocol interface and class of service are grouped into a flow. Traffic flow is summarized data that provides a simple, effective, and scalable way to gain visibility into traffic types and bandwidth usage on the network. One important fact about network flows is that flows do not provide any packet payload. Rather, only meta-information about network connections is collected. The meta-information contains several attributes (e.g., the packets or bytes transferred in a flow). Unlike packet data approach, since network flows do not carry packet payload, all information which was transported in the original payload is irretrievably lost. While the lack of payload contributed to some advantages such as privacy and scalability [13].

Based on the available information from the literature it seems that an efficient attack classification can be done by using the network traffic flow information. Recent researches showed that network traffic flows could improve the accuracy of attack classification [13, 18]. Therefore, the network traffic flow method has been used in this work to monitor network behaviour. There are many advantages in using flow data instead of packet data. The major advantage comes from protecting the privacy and the confidentiality of the protected network as well as the reduced need of storage space for the data, since network flows requires a one tenth of the original packet-based data which is a huge difference. Network traffic flow provides abstract overview of the network state, performance and behaviour which are required to train the anomalies classifier engine.

Two approaches were used to select the relevant features from the network traffic. Initially, an analysis of what information the field of literature holds on this topic; then an evaluation of different attack scenarios and how they affect the network traffic behaviour have been prepared. The most suitable traffic flow features are selected from the feature spectrum based on the prior knowledge about the environment that the IDS is monitoring and the analysis of known attack types.

4.1.1 Packet Features Selection

After analysing the features from the attack scenarios point of view and what have been utilized in the literature, it seemed that the features used by [16] are very comparable to the features that should be monitored for each attack class. Therefore the features used by [16] were chosen as well as some other related features obtained from the attack scenarios analysis. The features to be monitored are listed in Table 2. The selected feature set containing statistical information that reflects the amount of change within each time interval.

As illustrated in Table 2, twenty five features have been selected to be monitored. The selected features will be represented as a vector of 25 elements, where each element represents its designated value. At this stage the extracted vectors will be defined as the symptoms vectors. To expound on the functionality of the packet features extractor, the functional model of the proposed system is shown in Figure 2.

4.2 Anomaly Classifier Engine (ACE)

The anomaly classifier engine is responsible for automatically classify the detected activity and determine the attack class, based on predefined set of patterns of known attack mechanisms that are defined in the CAPEC and CVE databases. The PFE monitors network traffic flow and extracts traffic flow features to generate alarm meta-information as a vector representing symptoms vector. The symptoms vector is then passed to the anomaly classifier engine that automatically determines the attack class. The development of ACE goes through two stages. First, the ACE is trained with several types of attack symptoms vectors. Then, when the training is completed, the ACE is ready to classify new incoming alarms automatically.

During training phase, a signature-based IDS is deployed next to the A-IDS such that the two systems monitor the exact network traffic as illustrated in Figure 3. Once the A-IDS generates an alarm the anomaly classifier engine learns the alarm class from the signature-based system. The strategy of alarm labelling process is as follow; if A-IDSs' reported activity did not trigger the S-IDS to generate an alarm it shall be considered as false alarm otherwise the classification engine will acknowledge S-IDS classification of the detected activity. Once the training phase is over, the proposed system enters the classification phase. During this phase, the packet header extractor actively extracts network traffic flow features of A-IDS reported activities and the anomaly classifier engine classifies the events based on the learnt classification model. The ACE includes the algorithm used to classify attacks; machine learning technologies have been used for classification process, to automatically and systematically classify attacks detected by an anomaly-based intrusion detection system. Machine learning can help to automate tasks and provide predictions where humans have difficulties to comprehend large amount of data. One major benefit of machine learning is the generalization ability, in which it has the ability of an algorithm to function accurately on new, unseen examples after having trained on a learning data set.

4.2.1 Machine Learning Algorithm Selection

The choice of which specific learning algorithm should be used is a critical step. The classifier's evaluation is most often based on classification accuracy (the percentage of correct classifications divided by the total number of events in the data set). There are various techniques available used to calculate a classifier's accuracy. One technique is to split the training set by using two-thirds for training and the other third for estimating perfor-

Table 2: Selected network traffic flow-based features (RD: Resource Depletion, NR: Network Reconnaissance, Spf: Spoofing, ExA: Exploitation of Authentication, ExP: Exploitation of Privilege/Trust)

Label	Feature	RD	NR	Spf	ExA	ExP
F1	Number of sequences received during the observation period	✓	✓			
F2	Amount of bytes received during the observation period	✓				
F3	Total number of packet received	✓				
F4	Total number of sequences received during the observation period from different IP's	✓	✓	✓	✓	✓
F5	Number of sequences sent during the observation period	✓	✓			
F6	Amount of bytes sent during the observation period	✓				
F7	Total number of packet sent	✓				
F8	Total number of sequences sent during the observation period to different IP's	✓	✓	✓		✓
F9	Number of privileged port numbers used during the observation period			✓	✓	✓
F10	Number of different privileged port numbers used during the observation period			✓	✓	✓
F11	Number of registered ports used during the observation period			✓	✓	✓
F12	Number of different registered port numbers used			✓	✓	✓
F13	Total number of different TCP and UDP port numbers used by source	✓	✓	✓		
F14	Total number of different TCP and UDP port numbers used by the host	✓	✓	✓		
F15	Number of TCP requests for transmission	✓		✓		
F16	Number of half open connections	✓	✓	✓	✓	✓
F17	Number of established connections which represents an open connection	✓		✓		
F18	Number of connection termination requests sent	✓	✓	✓		
F19	Number of confirming connection termination received	✓	✓	✓		
F20	Total number of TCP connections during the observation period	✓	✓		✓	✓
F21	Total number of UDP flows during the observation period	✓	✓		✓	✓
F22	Total number of TCP connections initiated by source	✓	✓		✓	
F23	Total number of UDP flows received	✓	✓		✓	
F24	Total number of TCP connections initiated by the host	✓	✓		✓	
F25	Total number of UDP flows sent	✓	✓		✓	

mance. In another technique, known as cross-validation, the training set is divided into mutually exclusive and equal-sized subsets and for each subset the classifier is trained on the union of all the other subsets. The average of the error rate of each subset is therefore an estimate of the error rate of the classifier. If the error rate evaluation is unsatisfactory, the selected features must be re-examined.

Since the attack class and the related meta-information can be obtained, only supervised machine learning algo-

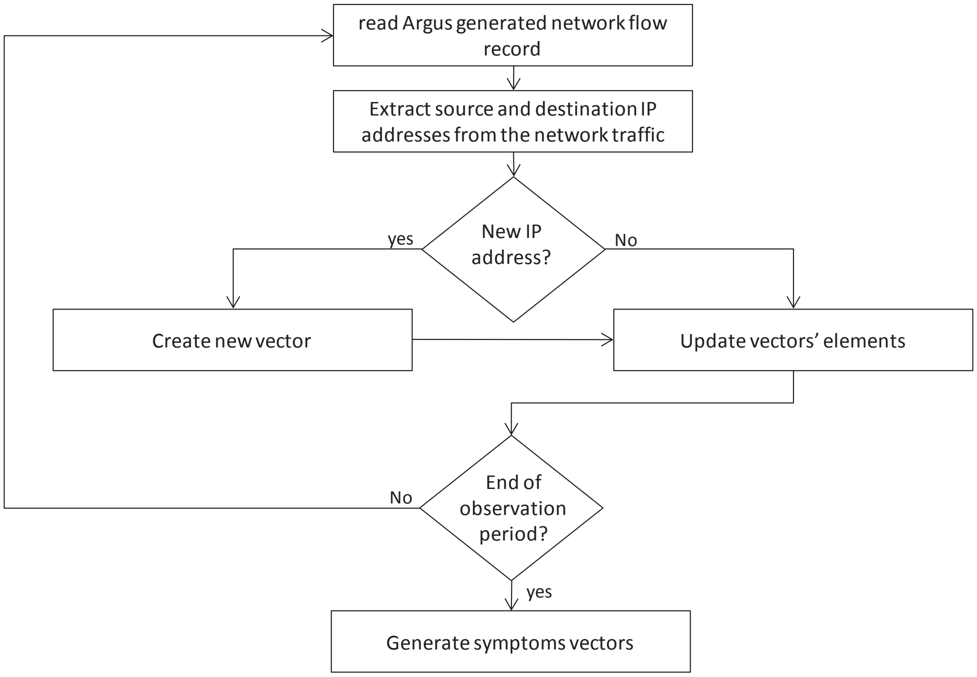
read Argus generated network flow record
Extract source and destination IP addresses from the network traffic
New IP address?
yes
No
Create new vector
Update vectors' elements
End of observation period?
No
yes
Generate symptoms vectors

Figure 2: Functional model of the proposed method

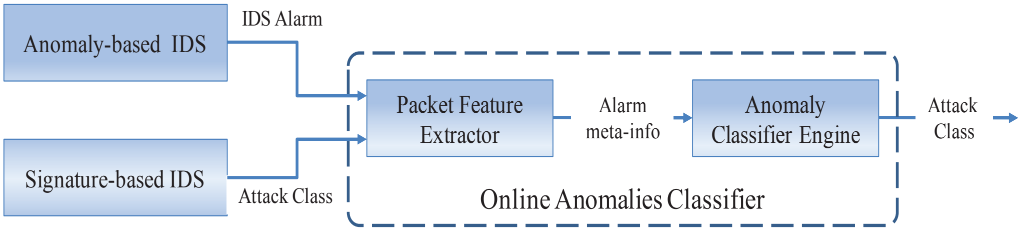

Figure 3: Online anomalies classifier during training phase

rithms have been considered in this work. These algorithms generally achieve better results than unsupervised methods. However, the classification algorithm must meet several requirements as listed in Table 3.

In this work, five machine learning algorithms have been considered as follows; Random Committee, Rotation Forest, PART, Random Forest and Random Tree. These algorithms implement supervised techniques, their training and classification phase are fast and able to handle large amount of data. In this work, machine learning evaluations have been implemented by using Weka platform. Weka is a well-known collection of machine learning algorithms, it also provide a comprehensive framework to execute benchmarks on several datasets under the same testing conditions.

Random Tree is a decision tree that considers number of randomly chosen attributes at each node. Random Tree have been introduced by [2] as a base classifier for his random forest classification algorithm. Random Tree develops un-pruned decision trees furthermore, it does not perform and optimization on its resultant rulesets.

Random Committee is an ensemble of randomized Random Tree classifiers. Each Random Tree classifier is built using a different random number seed. The final prediction is a straight average of the predictions generated by the individual base classifiers. Rotation Forest [32] have proposed an ensemble-classifier based on feature extraction. The model uses decision tree algorithms (J48) as base classifier and the feature extraction is based on Principal Component Analysis (PCA). PCA have been used to determine features feasibility and find out whether they do contribute to increased classification accuracy. In generating the training dataset, the feature set is randomly split into number of subsets and the Principal Component Analysis (PCA) is applied to each subset. The coefficients of the principal components is represented in a vector for each subset, and organized in a rotation matrix. All principal components are retained in order to preserve

Table 3: Machine learning selection criteria

Num.	Criteria	Description
1	Support for multiple classes	The attacks fall into five different categories. Therefore, it is required that, the selected algorithm supports multiclass classification.
2	Able to handle large amount of data	Using large amounts of memory can seriously degrade the system. Quite a few learning algorithms can be trained incrementally, one data row at a time. These methods generally have runtime that is linear in the number of rows and fields in the data and only require the current data row to be presented in the main memory. Because of this, they can process large amount of data.
3	High accuracy classification	One of the significant requirements is that, the machine learning algorithm should classify with high accuracy and low false positive and negative.
4	Able to train with small data set (fast training)	It is required that, the machine learning algorithm is able to develop the classification model in a small number of data set, to decrease the amount of alarms required.
5	Having an explicit underlying probability model	The machine learning algorithm should be based on statistical approaches, which provides a probability that an instance belongs in each class, rather than simply a classification.
6	Developed for academic researches	Because machine learning is beyond the scope of this work.

the variability information in the data. Thus, number of axis rotations takes place to form the new features for a base classifier. The proposed rotation forest ensemble have been evaluated on a selection of 33 benchmark data sets from the UCI repository and compared it with Bagging, AdaBoost, and Random Forest. The classification accuracy was more accurate than in AdaBoost and Random Forest, and more diverse than these in Bagging as well.

PART [6] have introduced PART rule-induction algorithm which utilized C4.5 and RIPPER rule-learning algorithms to propose a classification technique for inferring rules by repeatedly generating partial decision trees without the needs for complex optimization. It adapts the separate-and-conquer strategy in that it builds a rule, removes the instances it covers and continues creating rules recursively for the remaining instances until none are left. In essence to make a single rule, a decision tree is build for a selected set of instances, then the leaf with the largest coverage is made into a rule and that decision tree will be discarded. PART is a partial decision tree algorithm, which is the developed version of C4.5 and RIPPER algorithms. The main speciality of the PART algorithm is that it does not need to perform global optimisation like C4.5 and RIPPER to produce the appropriate rules; instead it utilises separate-and-conquer methodology to builds a partial C4.5 decision tree recursively and makes the “best” leaf into a rule.

Random Forests; is a combination of decision trees such that each constructed tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest. The concept behind the random forests is that, significant improvements in classification accuracy would achieve from growing an ensemble of trees furthermore each tree to vote for the most popular class. Random forests have been introduces by [2] and have been defined as an ensemble learning method for classification that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

5 Evaluation Results of NAC

This section presents the evaluation results of the proposed network anomaly classifier. First, it describes the dataset employed and then the evaluation results are presented.

Table 4: Machine learning selection criteria

Attack Class	Dataset A	Dataset B	Dataset C	Dataset D	Dataset E
dos	615	1004	689	0	1378
u2r	15	0	1807	1084	3614
r2l	310	333	148	0	270
data	41	0	2	198	114
probe	171	13	144	148	195
Total	1152	1350	2790	1430	5571

5.1 Evaluation Dataset

The selected machine learning algorithms have been evaluated against five different datasets. The evaluation was based on the classification accuracy using the defined network traffic features. The datasets contain network traffic features representing network state during alarms identified by security analyst or raised by signature-based IDS (attack only dataset); each dataset contains number of instances representing network traffic audit records during a detected malicious activity as shown in Table 5 and Figure 4 illustrate the percentage distribution of attack types in datasets

Dataset A: This dataset contains 1152 instances, having majority of denial of service attacks by random selection. The occupancy ratio of denial of service attacks and remote to local attacks is nearly 2:1, and the ratio of remote to local attacks and probe is also about 2:1. The dataset contains some attacks representing the user to root and data attacks. However, some classes have few audit records, which may impact negatively to the detection accuracy.

Dataset B: contains 1350 instances, having majority of dos attacks and some other attacks randomly selected, this dataset represents a scenario when an attacker uses probe and remote to user attacks to cause network resource unavailable to its intended users, which is common in real scenarios.

Dataset C: include 2790 instances represents a scenario when an attacker uses probe and remote to user attacks with dos to gain root privileges. Therefore, the dataset have a majority of user to root attacks. The occupancy ratio of denial of service attacks and remote to local attacks is nearly 1:2.

Dataset D: include 1430 instances represents the same scenario of Dataset C when an attacker uses probe and remote to user attacks to gain root privileges but without the using of dos attacks.

Dataset E: This dataset contains 5571 instances randomly collected, having a majority of u2r attacks. The occupancy ratio of denial of service attacks and remote to local attacks is nearly 2:1.

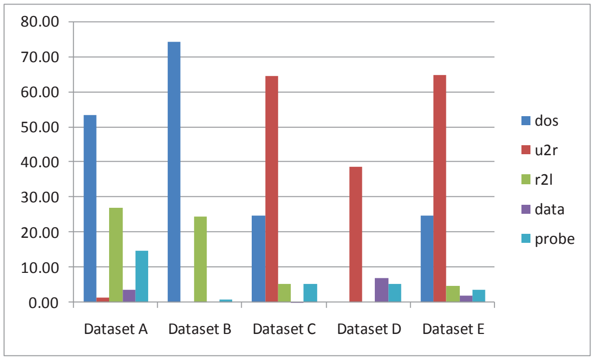

Figure 4: Percentage distributions of attack types in datasets

5.2 Evaluation Result

Three performance metrics have been used for machine learning comparison, classification accuracy, Precision and F-Measure. The performance of the selected machine learning algorithms have been conducted by training and testing with above five datasets to show its performance in different scenarios. However, there are four factors which influence the classification accuracy; the number of samples (alarms) processed during training phase, the frequency distribution of the alarms, the machine learning used and the network traffic features used. Table 4 illustrates the detection accuracy of the five datasets using different machine learning algorithms. Based on the above results of five datasets, it can conduct that Random Committee and Random Tree perform better than other algorithms and their detection accuracy almost identical, but the precision of Random Committee is higher than Random Tree. Therefore, in this work Random Committee will be used to classify the detected activities.

6 Conclusions and Future Works

In order to estimate the risk of unknown attacks, a solution to automate the classification of anomaly-based alarms is required. However, So far no effective and efficient automatic or semi-automatic approach is currently available able to classify anomaly-based alarms at run-

Table 5: Machine learning selection criteria

Machine Learning	**Dataset A**	**Dataset B**	**Dataset C**	**Dataset D**	**Dataset E**
Random Committee	96.78%	99.85%	98.49%	99.23%	98.20%
Rotation Forest	94.18%	99.03%	97.88%	98.04%	98.09%
PART	93.22%	99.18%	97.13%	98.04%	98.06%
Random Forest	96.61%	99.70%	98.45%	99.09%	98.18%
Random Tree	96.78%	99.85%	98.49%	99.23%	98.20%

time. Thus, any anomaly-based alarm must be manually processed to identify its class; this may increase the workload of security analyst, and will effectively increase time required; as well as, the dependence on security analysts.

This paper presents Network Anomalies Classifier (NAC) that uses machine learning technologies to automatically classify activities detected by a packet header-based anomaly detection system. The concept behind the proposed methodology is that, attacks those share some common network traffic flow behaviors are usually in the same class. Based on the available information from the literature it seems that an efficient attack classification can be done by using the network traffic flow information. Recent researches showed that network traffic flows could improve the accuracy of attack classification. Therefore, the network traffic flow method has been used in this work to monitor network behaviour. Thus by extracting traffic flow sequences triggered by certain attack, it is possible to compare those sequences to previously collected data using machine learning algorithm, then to infer the attack class from the matching sequences.

Two approaches were used to select the relevant features from the network traffic. Initially, an analysis of what information the field of literature holds on this topic; then an evaluation of different attack scenarios and how they affect the network traffic behaviour have been prepared. The most suitable traffic flow features are selected by handpicking from the feature spectrum based on the prior knowledge about the environment that the IDS is monitoring and the analysis of known attack types.

In this work, five machine learning algorithms have been considered as follows; Random Committee, Rotation Forest, PART, Random Forest and Random Tree. Evaluation experiments showed that machine learning algorithms are capable of classifying malicious activities in an effective and efficient means. However, a too low number of samples could generate an inaccurate classification. Therefore, as the number of training samples increases, accuracy increases. Based on the evaluation experiments results, it can conduct that Random Committee and Random Tree perform better than other algorithms and their detection accuracy almost identical, but the precision of Random Committee is higher than Random Tree. Therefore, as future works random committee algorithm will be used to classify the detected activities to estimate the security risk level.

References

[1] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *International Journal of Network Security*, vol. 18, no. 3, pp. 420–432, 2016.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] C. Y. Chiu, Y. J. Lee, C. C. Chang, W. Y. Luo, and H. C. Huang, "Semi-supervised learning for false alarm reduction," in *Advances in Data Mining Applications and Theoretical Aspects,* Springer Berlin Heidelberg, vol. 6171, pp. 595–605, 2010.

[4] G. Fernandes and P. Owezarski, "Automated classification of network traffic anomalies," *Security and Privacy in Communication Networks*, vol. 19, pp. 91–100, 2009.

[5] R. Fontugne, T. Hirotsu, and K. Fukuda, "An image processing approach to traffic anomaly detection," in *ACM Proceedings of the 4th Asian Conference on Internet Engineering (Aintec'08)*, pp. 17, 2008.

[6] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 144–151, 1998.

[7] V. N. L. Franqueira, Z. Bakalova, T. T. Tun, and M. Daneva, "Towards agile security risk management in RE and beyond," in *IEEE Workshop on Empirical Requirements Engineering*, pp. 33–36, 2011.

[8] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, and A. Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks: A survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 271, 2013.

[9] A. A. Ghorbani, W. Lu, and M. Tavallaee, "Network intrusion detection and prevention," *Information Security*, vol. 47, pp. 27–54, 2010.

[10] M. Guimaraes and M. Murray, "Overview of intrusion detection and intrusion prevention," in *Proceedings of the ACM 5th Annual Conference on Information Security Curriculum Development,* pp. 44–46, 2008.

[11] X. Hongbin and X. Wenbo, "Research on method of network abnormal detection based on hurst parameter estimation," in *Proceedings International Confer-*

ence on Computer Science and Software Engineering, vol. 3, pp. 559–562, 2008.

[12] G. Javadzadeh and R. Azmi, "IDuFG: Introducing an intrusion detection using hybrid fuzzy genetic approach," *International Journal of Network Security Its Applications*, vol. 17, no. 6, pp. 754–770, 2015.

[13] J. H. Jun, D. Lee, C. W. Ahn, and S. H. Kim, "DDoS attack detection using flow entropy and packet sampling on huge networks," in *The Thirteenth International Conference on Networks*, pp. 185–190, Nice, France, 2014.

[14] Y. Kim, J. Y. Jo, and K. K. Suh3, "Baseline profile stability for network anomaly detection," *International Journal of Network Securit*, vol. 6, no. 1, pp. 60–66, 2008.

[15] J. M. Kizza, "Introduction to computer network vulnerabilities," in *Guide to Computer Network Security*, vol. 4, pp. 87–103, 2015.

[16] O. Knuuti, T. Seppälä, T. Alapaholuoma, J. Ylinen, P. Loula, P. Kurnpulainen, and K. Hätönen, "Constructing communication profiles by clustering selected network traffic attributes," in *5th International Conference on Internet Monitoring and Protection (ICIMP'10)*, pp. 105–109, 2010.

[17] P. G. Kumar and D. Devaraj, "Network intrusion detection using hybrid neural networks," in *2007 International Conference on Signal Processing Communications and Networking*, pp. 563–569, 2007.

[18] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 217, 2005. Press.

[19] F. Y. Leu and Z. Y. Li, "Detecting DoS and DDoS attacks by using an intrusion detection and remote prevention system," in *Fifth International Conference on Information Assurance and Security*, pp. 251–254, 2009.

[20] H. H. Lin, C. H. Mao, and H. M. Lee, "False alarm reduction by weighted score-based rule adaptation through expert feedback," in *IEEE 2nd International Conference on Computer Science and its Applications*, pp. 1–8, 2009.

[21] T. Liu, Z. Wang, H. Wang, and K. Lu, "An entropy-based method for attack detection in large scale network," *International Journal of Computer, Communications and Control*, vol. 7, no. 3, pp. 242–250, 2012.

[22] M. V. Mahoney, "Network traffic anomaly detection based on packet bytes," in *Proceedings of the ACM Symposium on Applied Computing* , pp. 346, 2003.

[23] Y. Meng and L. Kwok, "Adaptive false alarm filter using machine learning in intrusion detection," *Practical Applications of Intelligent Systems*, pp. 573–584, 2011.

[24] Mitre Corporation, "Common attack pattern enumeration and classification (CAPEC)," 2011.

[25] N. Mohd, S. Annapurna, and H. S. Bhadauria, "Taxonomy on security attacks on self configurable networks," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 44–52, 2015.

[26] T. L. Nielsen, J. Abildskov, P. M. Harper, I. Papaeconomou, and R. Gani, "The CAPEC Database," *Journal of Chemical & Engineering Data*, vol. 46, pp. 1041–1044, 2001.

[27] A. Niemelä, "Traffic Analysis for Intrusion Detection in Telecommunications Networks," *Master of Science Thesis, Tampere University of Technology,* 2011.

[28] I. V. Onut and A. A. Ghorbani, "A feature classification scheme for network intrusion detection," *International Journal of Network Security*, vol. 5, no. 1, pp. 1–15, 2007.

[29] S. Parsazad, E. Saboori, and A. Allahyar, "Fast feature reduction in intrusion detection datasets," in *MIPRO Proceedings of the IEEE 35th International Convention*, pp. 1023–1029, 2012.

[30] Q. Qian, T. Wang, and R. Zhan, "Relative network entropy based clustering algorithm for intrusion detection," *International Journal of Network Security*, vol. 15, no. 1, pp. 16–22, 2013.

[31] O. Rodas and M. A. To, "A study on network security monitoring for the hybrid classification-based intrusion prevention systems," *International Journal of Space-Based and Situated Computing*, vol. 5, no. 2, pp. 115, 2015.

[32] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1619–30, 2006.

[33] N. Sharma and S. Mukherjee, "A layered approach to enhance detection of novel attacks in IDS," *International Journal of Advances in Engineering Technology*, vol. 4, no. 2, pp. 444–455, 2012.

[34] H. Wang, M. Guo, L. Zhou, and J. Camargo, "Ranking attacks based on vulnerability analysis," in *2010 43rd IEEE Hawaii International Conference on System Sciences*, pp. 1–10, 2010.

[35] K. Wang, S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, LNCS 3224, pp. 203–222, Springer, 2004.

[36] J. Yu and Y.V.R. Reddy, "TRINETR: an intrusion detection alert management systems," in *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 235–240, 2004.

Qais Saif Qassim is a Ph.D. candidate in Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM). His research interest in network security and management.

Abdullah Mohd Zin received his PhD from the University of Nottingham, United Kingdom in 1993. He is currently the dean of Faculty of Information Science and Technology, UKM. His specialization in system architecture, programming language, communication and distributed, formal method.

Mohd Juzaiddin Ab Aziz received his PhD degrees in Computer Science from University Putra Malaysia (UPM). Currently, he is the deputy dean of undergraduate studies in faculty science and information technology, UKM. His specialization in programming language technology and natural language processing.