The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems

Conference Paper · November 2015

DOI: 10.1109/BADGERS.2015.014

CITATIONS
20

READS
1,229

2 authors:

Nour Moustafa
UNSW Canberra
45 PUBLICATIONS **418** CITATIONS

SEE PROFILE

Jill Slay
La Trobe University
145 PUBLICATIONS **1,287** CITATIONS

SEE PROFILE

2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security

The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems

Nour Moustafa, Jill Slay

School of Engineering and Information Technology
University of New South Wales at the Australian Defence Force Academy
nour.abdelhameed@student.adfa.edu.au, j.slay@adfa.edu.au

***Abstract*—Because of the increase flow of network traffic and its significance to the provision of ubiquitous services, cyber-attacks attempt to compromise the security principles of confidentiality, integrity and availability. A Network Intrusion Detection System (NIDS) monitors and detects cyber-attack patterns over networking environments. Network packets consist of a wide variety of features which negatively affects detection of anomalies. These features include some irrelevant or redundant features which reduce the efficiency of detecting attacks, and increase False Alarm Rate (FAR). In this paper, the feature characteristics of the UNSW-NB15 and KDD99 datasets are examined, and the features of the UNSW-NB15 are replicated to the KDD99 data set to measure their effeciency. we apply An Association Rule Mining algorithm as feature selection to generate the strongest features from the two data sets. Some existing classifiers are utilised to evaluate the complexity in terms of accuracy and FAR. The experimental results show that, the original KDD99 attributes are less efficient than the replicated UNSW-NB15 attributes of the KDD99 data set. However, comparing the two data sets, the accuracy of the KDD99 dataset is better than the UNSW-NB 15 dataset, and the FAR of the KDD99 dataset is lower the UNSW-NB 15 dataset.**

***Index Terms*—UNSW-NB15 dataset, KDD99 dataset, Feature extraction, Feature selection, Network Intrusion Detection System (NIDS)**

I. Introduction

As networks are considered as the engine of communications, attackers endeavour to penetrate them to steal valuable information or disrupt computer resources. A Network Intrusion Detection System (NIDS) is technique to protect computer resources against malicious activities [1]. NIDS methodologies can be categorised as Misuse Detection (MD) and Anomaly Detection (AD) [2]–[4]. MD uses patterns or signatures of existing attacks to detect known intrusions. However, AD establishes a normal profile of activities, and any strong deviations from this profile are reflected as an attack. Alghouth, The FAR of AD is high, AD is able to detect novel attacks. Therefore, many studies have suggested its use [5]–[7].

To decrease FAR, the construction of NIDS needs to extract and choose the relevance features of raw network traffic. Feature extraction captures attributes from network packets. Some of these attributes are redundant or irrelevant; thus reducing the accuracy of detection. Feature selection, on the other hand, removes redundant and noisy attributes from high dimensional data sets and selects a subset of relevant attributes to establish a reliable NIDS model [8].

An association rule mining (ARM) technique generates feature correlations from a data set, as it can find out related isomorphism between data set observations [9], [10]. The association rule mechanisms were applied to extract suitable behaviors from user activities. The mined rules of audit data are integrated and combined into an aggregate rule set to represent normal profile of users. To analyse user login sessions, frequent patterns were extracted from the sequence of commands during the session and new patterns were compared with the established profile patterns [10]. In this paper, we suggest a new algorithm based on ARM as a feature selection method to adopt the relevent features from the UNSW-NB15 and the KDD-99 data sets.

The UNSW-NB15 data set has recently been released [11]. This data set contains nine different modern attack types and wide varieties of real normal activities. The simulation testbed configuration depended on generating network traffic with the change over time to imitate the contemporary real network traffic. This data set consists of 49 features with the class label that involves characteristics of the network traffic using the flow based between hosts (i.e., client-to-server or server-to-client) and the packet header [12].

The complexity of the UNSW-NB15 data set was evaluated by customising a part from this data set as a training set and a testing set in three aspects, and compared it with the benchmark data set of the KDD99. The first aspect estimated the statistical analysis of the training and the testing set using the Kolmogorov-Smirnov Test, skewness, and kurtosis approaches. Secondly, we examined the feature correlations, first, with the class label using a Pearson's Correlation Coefficient function, and second, without the class label utilising a Gain Ratio method. Third, the complexity in terms of accuracy and FAR were assessed utilising some existing classification techniques. The results showed that the training and testing set cannot be represented under the normal probability distribution. The extracted features were almost correlated, whether

978-1-4673-8944-0/16 $31.00 © 2016 IEEE
DOI 10.1109/BADGERS.2015.14

with the class label or without. The complexity of the two data sets demonstrated that the UNSW-NB15 data set is more complex than the KDD99 data set [13]. As a result, New NIDS algorithms can be evaluated upon the UNSW-NB15 data set, due to containing it a wide variety of patterns that represent modern real network traffic.

In this paper, we evaluate the complexity of the two data sets by replicating the same attributes of the UNSW-NB15 to the second week of the KDD99 data set. ARM is utilised as feature selection to choose the highest ranked features of the two data sets. The Naïve Bayes and EM clustering techniques are applied to appraise the accuracy and the FAR of the two data sets with both the original KDD99 attributes and the UNSW-NB15 attributes.

The contibution of this paper can be summarised as follows:

- We develop a feature selection method based on an Association Rule Mining (ARM) technique to adopt the best ranked attributes in the UNSW-NB15 and KDD99 data sets.
- The attributes of the UNSW-NB15 data set is replicated to the KDD-99 data set for computing the efficiency of the UNSW-NB15 features.
- The EM clustering and Naïve Bayes algorithms are applied on the best features of the two data sets to estimate the efficiency in terms of accuracy and False Alarm Rate (FAR).

The remainder of the paper is organised as follows. Section II discusses related work and background. Section III describes the KDD99 data set, while Section IV describes the UNSW-NB15 data set. In Section V, the proposed methodology is elaborated utilising three layers: network layer, processing layer and evaluation layer. The experimental results are presented in Section VI. Finally, Section VII provides the conclusion of the paper and examines directions for future research.

II. Related Work and Background

The aim of ARM is to generate the strongest itemsets among features by estimating support and confidence of each rule in a data set [9], [10], [14]–[16]. Several researchers have utilised ARM approaches in NIDSs. Agrawal et al. [10] stated that program executions and user activities show frequent correlations between system features. Lee et al. [14]utilised the basic notions of ARM to mine rules for system audit data in order to construct a normal user profile. Any deviation from the constructed profile constitutes missing or adding new rules, with respect to a violation of the rules (i.e., same antecedent, but different consequent), and significant changes in support of the rules. However in both these studies, the processing time of the ARM is very high. Luo et al. [15] designed an integrated fuzzy logic and ARM NIDS. These authors classified the quantitative features of the audit data into categories that have fuzzy membership values (i.e., Low, Medium and High). However, the customising of fuzzy membership function generates high FAR. Shi et al. [17] extended the work of membership values utilising a Genetic Algorithm to optimise the fuzzy-membership function parameters.

Zhang et al. [16] developed a partition-based ARM algorithm. The algorithm was configured to scan twice the training set. During the first scan, the data set was partitioned to execute easily into memory, while during the second scan, the item sets of the training set were generated. However, the complexity of this algorithm is very high. Similarly, Ming-Yang et al. [18] suggested an incremental fuzzy ARM model. They used a linked-list approach to store all candidate itemsets and their support into memory. The main disadvantage of this algorithm is a massive main memory requirement to store all candidate itemsets of every size. Ratchadaporn et al. [19] proposed probability-based incremental ARM algorithm. Nevertheless, the authors assumed that statistics of recently added records are not similar to old database records and a large number of new records were inserted into a data set. Nath et al. [20] stated a review of some existing dimensionality reduction techniques based on ARM methods. Some of these algorithms support single objective and others multi-objective. The results showed that the multi-objective ARM can be used to solve several real datasets. This study is related to our work in customising ARM as feature selection.

III. Description of the KKD 99 data set

The IST group of Lincoln laboratories at MIT University performed a simulation with normal and attack traffic in a military network (U.S. Air Force LAN) environment to generate the first version of the KDD99, namely DARAP98. This data set contains nine weeks from the simulation in raw tcpdump files. This data set was divided into a training set and a testing set. The training set was around 4 GBs from seven weeks of the simulated network traffic processed into 5 million connection records. Conversely, the testing set was two weeks and contained 2 million connection records.

The DARPA98 data set was configured to extract the 41 features for each vector with the class label using Bro-IDS tool, and was called KDD99 data set. In the KDD99 data set, the extracted features were divided into three groups: intrinsic features, content features and traffic features. The attack records of this data set were classified into four vectors: DoS, Probe, U2R, and R2L. The training set of KDD99 included 22 attack types and the testing set contained 15 attack types [12], [21].

There was also an upgraded version from the KDD99 data set, named NSLKDD [22]. In the NSLKDD data set, three major challenges were solved. First, the duplication of the records in the training set and test set was removed to eliminate from biasing classification systems towards the most repeated records. Second, the training set and testing set were generated by choosing a variety of the records from different parts of the original KDD99 data set to achieve authentic results while applying classification systems. Then, the unbalanced problem between the number of the training and testing was addressed to reduce the FAR.

All these versions of the data set are still applied to evaluate NIDSs, due to their public availability. However, many researchers have stated three major disadvantages [23]–[26] which can affect the trust of NIDSs evaluation. Firstly, attack data packets have a time to live value (TTL) of 126 or 253, whilst the packets of the network traffic mostly have a TTL of 127 or 254. However, TTL values of 126 and 253 do not happen in the training vectors of the attack types [23]. Secondly, the probability distribution of the testing set is different from the probability distribution of the training set, because of inserting new attack vectors in the testing set [24], [25]. This leads to skew or bias classification methods towards some records rather than balance between the attack and normal vectors. Thirdly, the data set is outdated; hence, it does not a full representation of modern normal and attack activities [26].

IV. Description of the UNSW-NB15 data set

The UNSW-NB 15 data set was created by utilising an IXIA PerfectStorm tool to extract a hybrid of modern normal and contemporary attack activities of network traffic. A tcpdump tool was used to capture 100 GB of raw network traffic (pcap files). Each pcap file contains 1000 MB in order to make analysis of packets easier. Argus and Bro-IDS techniques and twelve procedures were executed in a parallel implementation to generate 49 features with the class label. This data set contains 2, 540,044 records which are stored in four CSV files. Moreover, a part from this data set was divided into a training set and a testing set. The training set involved 175,341 records, while the testing set contained 82,332 records from all involved attack types and normal records [11], [12].

The involved attacks of the UNSW-NB15 data set were categorised into nine types as in the following points:

1) **Fuzzers**: is an attack in which the attacker attempts to discover security loopholes in an application, operating system or a network by feeding it with the massive inputting of random data to make it crash.
2) **Analysis**: is a type of variety intrusions that penetrate the web applications via ports (e.g. port scans), emails (e.g. spam) and web scripts (e.g. HTML files).
3) **Backdoor**: is a technique of bypassing a stealthy normal authentication, securing unauthorised remote access to a device, locating the entrance to plain text, as it struggles to continue unobserved.
4) **DoS**: is an intrusion which disrupts the computer resources via memory so as to cause excessive business, in order to prevent authorised requests from accessing a device.
5) **Exploit**: is a sequence of instructions that takes advantage of a glitch, bug or vulnerability, causing an unintentional or unsuspected behavior on a host or a network.
6) **Generic**: is a technique that establishes against every block-cipher using a hash function to cause a collision without respect to the configuration of the block-cipher.
7) **Reconnaissance**: also can be defined as a probe, and it is an attack which gathers information about a computer network to evade its security controls.
8) **Shellcode**: is malware in which the attacker penetrates a slight piece of code starting from a shell to control the compromised machine.
9) **Worm**:is an attack in which the attacker replicates itself to spread on other computers. Often, it utilises a computer network to spread itself, depending on the security failures of the target computer used to access it.

The involved features of the UNSW-NB 15 data set are classified into six groups as follows and reflected in Table I.

1) **Flow features**: this group includes the identifier attributes between hosts, such as client-to-serve or server-to-client.
2) **Basic features**: this category involves the attributes that represent protocols connections.
3) **Content features**: this group encapsulates the attributes of TCP/IP; also they contain some attributes of http services.
4) **Time features**: this category contains the attributes of time, for example, arrival time between packets, start/end packet time and round trip time of TCP protocol.
5) **Additional generated features**: this category can be further divided into two groups: (1) General purpose features (from number 36 - 40) which each feature has its own purpose, in order to protect the service of protocols. (2) Connection features (from number 41- 47) are built from the flow of 100 record connections based on the sequential order of the last time feature.
6) **Labelled Features**: this group represents the label of each record.

V. Proposed Methodology

In this section, we describe the proposed methodology to compare the efficiency and reliability of the KDD99 and the UNSW-NB data sets. The architecture consists of three layers: network layer, processing layer and evaluation layer, as shown in Figure 1.

A. The network layer

This layer represents network packets that were used to extract and generate the features. Network packets were analysed based on the flow between hosts and packet headers to recognise features that can discriminate between normal and attack activities [12].

B. The processing layer

This layer consists of three components: Extracted and Generated features, Feature selection and Decision engine. The Extracted and Generated features are discussed in Section IV, while Feature selection and Decision engine are described as follows:

Feature selection is utilised by an association Rule Mining (ARM) [10], [27] which is a data mining method to estimate the correlation of two or more than two features in a data set, since it can find the strongest itemsets between records. To

TABLE I: The Features of the UNSW-NB15 data set

#	Name	Description
1. Flow Features		
1	srcip	Source IP address.
2	sport	Source port number.
3	dstip	Destinations IP address.
4	dsport	Destination port number.
5	proto	Protocol type, such as TCP, UDP.
2. Basic Features		
6	state	The states and its dependent protocol e.g., CON.
7	dur	Row total duration.
8	sbytes	Source to destination bytes.
9	dbytes	Destination to source bytes.
10	sttl	Source to destination time to live.
11	dttl	Destination to source time to live.
12	sloss	Source packets retransmitted or dropped.
13	dloss	Destination packets retransmitted or dropped.
14	service	Such as http, ftp, smtp, ssh, dns and ftp-data.
15	sload	Source bits per second.
16	dload	Destination bits per second.
17	spkts	Source to destination packet count.
18	dpkts	Destination to source packet count.
3. Content Features		
19	swin	Source TCP window advertisement value.
20	dwin	Destination TCP window advertisement value.
21	Stcpb	Source TCP base sequence number.
22	dtcpb	Destination TCP base sequence number.
23	smeansz	Mean of the packet size transmitted by the srcip.
24	dmeansz	Mean of the packet size transmitted by the dstip.
25	trans_depth	The connection of http request/response transaction.
26	res_bdy_len	The content size of the data transferred from http.
4. Time Features		
27	sjit	Source jitter.
28	djit	Destination jitter.
29	stime	Row start time.
30	ltime	Row last time.
31	sintpkt	Source inter-packet arrival time.
32	dintpkt	Destination inter-packet arrival time.
33	tcprtt	Setup round-trip time, the sum of 'synack' and 'ackdat'.
34	synack	The time between the SYN and the SYN_ACK packets.
35	ackdat	The time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	If srcip (1) = dstip (3) and sport (2) = dsport (4), assign 1 else 0.
5. Additional Generated Features		
37	ct_state_ttl	No. of each state (6) according to values of sttl (10) and dttl (11).
38	ct_flw_http_mthd	No. of methods such as Get and Post in http service.
39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	No of flows that has a command in ftp session.
41	ct_srv_src	No. of rows of the same service (14) and srcip (1) in 100 rows.
42	ct_srv_dst	No. of rows of the same service (14) and dstip (3) in 100 rows.
43	ct_dst_ltm	No. of rows of the same dstip (3) in 100 rows.
44	ct_src_ ltm	No. of rows of the srcip (1) in 100 rows.
45	ct_src_dport_ltm	No of rows of the same srcip (1) and the dsport (4) in 100 rows.
46	ct_dst_sport_ltm	No of rows of the same dstip (3) and the sport (2) in 100 rows.
47	ct_dst_src_ltm	No of rows of the same srcip (1) and the dstip (3) in 100 records.
6. Labelled Features		
48	Attack_cat	The name of each attack category.
49	Label	0 for normal and 1 for attack records

define the ARM, let $r = \{f_1, f_2, f_3, ..., f_N\}$be a set of features and D be a data set consisting of T transactions $t_1, t_2, t_3, ...t_N$. Each transaction t_j, $\forall 1 \leq j \leq N$ is a set of features such that $t_j \subseteq r$. The association rule $f_1(antecdent) \Rightarrow f_2(precedent)$ subjects to the constraints of (1) $\exists t_j, f_1, f_2 \in t_j$ (2) $f_1 \subset r, f_2 \subset r$, and (3) $f_1 \cap f_2 \in \Phi$.

The ARM depends on two measures: *support* and *confidence* to generate rules. *Support* determines the frequency

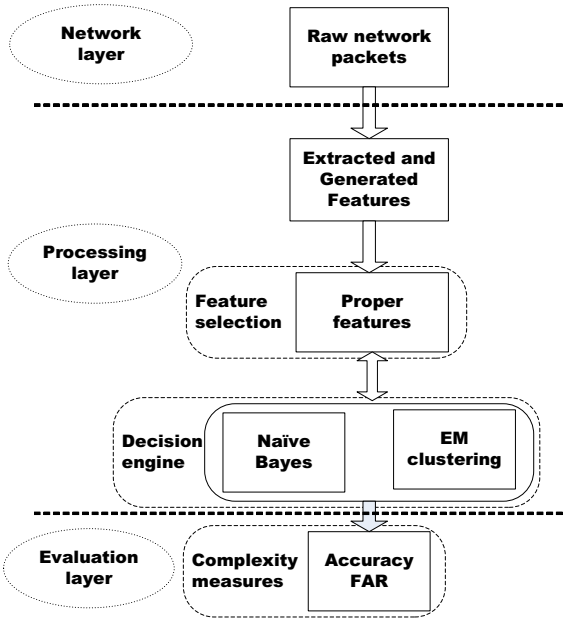

Fig. 1: The proposed architecture for comparing the KDD99 and the UNSW-NB15 data sets

of record values that represents the correlation percentage, as given in Equation (1). In Equation (2), *confidence* is the frequency of a precedent if the antecedent has already happened.

$$sup(f_1 \Rightarrow f_2) = \frac{\mid \#t_j \mid f_1, f_2 \in t_j \mid}{N} \quad (1)$$

$$conf(f_1 \Rightarrow f_2) = \frac{\mid \#t_j \mid f_1, f_2 \in t_j \mid}{\mid \#t_j \mid f_1 \in t_j \mid} \quad (2)$$

The ARM is generated to find all frequent itemsets, and identifying the strongest rules in the frequent itemsets. The strongest ARM in D are achieved, when support of a rule is greater than user-specified minimum support (i.e., $sup \geq minsup$), and confidence of a rule is greater than minimum confidence thresholds (i.e., $conf \geq minconf$).

Based on the above discussion, Apriori algorithm [27] was developed to be the first ARM algorithm. The procedures of applying Apriori algorithm is divided into two steps. The first step finds out iteratively all frequent itemsets which achieve $sup \geq minsup$. The second step establishes association rules that satisfy $conf \geq minconf$, as described in Algorithm 1.

We developed algorithm 2 to generate the best attributes based on the ARM. Line 1 implements Algorithm 1 to create all possible rules in a data set (e.g., KDD99 and UNSW-NB15). From line 2 to 13, check if the rules do not achieve the ARM constraints, remove it . Otherwise, compute support and confidence from Equations 1 and 2. In Line 12 and 13, the itemset of rules is updated with the computed values of

Algorithm 1 The procedures of implementing the Apriori algorithm [27]

Input: T transactions, α threshold of minsup, *minconf*

Output: L_K frequent itemset of size K

1: $L_1 = \{\text{large 1- itemsets}\}$
2: k = 2
3: **while** $(L_{K-1} \neq \Phi)$ **do**
4: C_K =Generate (L_{K-1})
5: **for** (each condidate $t \in T$) **do**
6: C_t =subset (C_K, t)
7: **for** (each condidate $c \in C_t$) **do**
8: *count [C] =count [C]+1*
9: **end for**
10: **end for**
11: $L_K = \{c \in C_K \mid count[C] \geq \alpha\}$
12: $K = K + 1$
13: **end while**
14: **return** $\cup_K L_K$

Algorithm 2 Feature selection based on the ARM

Input: *D* (training set), *minsup* (minimum support threshold), *minconf* (minimum confidence threshold) , *C* (class label), *X* (number of required features)

Output: *F* (feature subset)

1: *R* = Apriori(D, minsup, minconf, C)
2: **for** (i=1 to length(R)) **do**
3: **if** (R[i]==R[i+1]) **then**
4: count[i] = count[i] + 1
5: **else**
6: count[i]=1
7: **end if**
8: filter_R[i]= R- R[i]
9: **end for**
10: **for** (j=1 to length(filter_R)) **do**
11: **if** (count[j]<=1 || R[j] $\notin$C) **then**
12: sup[j] = count[j] / length(filter_R)
13: conf[j] = count[j] / length (D[j])
14: **end if**
15: **end for**
16: Sort (filter_R, sup, conf)
17: **for** (m=1 to X) **do**
18: **if** (sup >= minsup && conf >= minconf) **then**
19: F =F + (extracted_features (r) ,C)
20: **end if**
21: **end for**
22: **return** F

support and confidence. In Line 16, all rules order descending based on the values of support and confidence. From Line 17 to 22, the strongest features are constructed based on the number of required features, *sup* $\geq$ *minsup* and *conf* $\geq$ *minconf*. Figure 2 shows an example on the UNSW-NB15 to describe the executing of Algorithm 2.

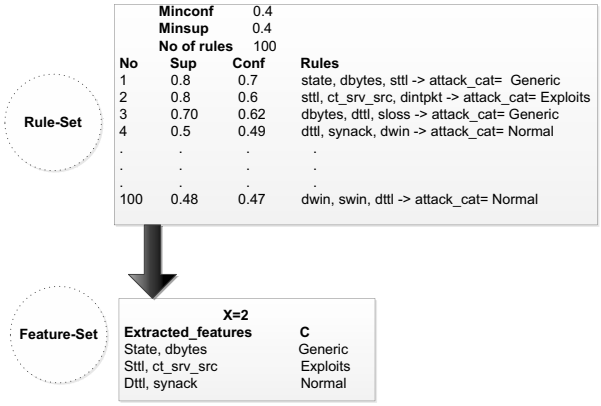

Fig. 2: An example of feature selection ARM on the UNSW-NB15 data set

In Decision engine, we applied Naïve Bayes (NB) [28] and EM clustering [29] models. The NB model is a conditional probability model which establishes the classification of the two classes (i.e., normal (0) or attack (1)). It is used by using the maximum a posterior (MAP) function which is denoted as:

$$P(C \mid I) = argmax_{w \in \{1,2,..,N\}} P(C_w) \Pi_{j=1}^{N} P(I_j \mid C_w) \quad (3)$$

such that C is the class label, *I* is the observation of each class, w is the class number, *P(C\I)* denotes the probability of the class given a specified observation and $P(C_w \Pi_{j=1}^{N} P(I_J \mid C_w))$ indicates indicates multiplication of all the probabilities of the instances conditionally to their classes to achieve the maximum outcome. The Expectation-Maximisation (EM) clustering technique depends on maximising the probability density function of a Gaussian distribution to calculate the mean and the covariance of each instance *I* in *T*. The EM clustering algorithm encompasses into two steps (i.e., Expectation (E-step) and Maximization (M)). In the E-step, the estimated likelihood for each instance I in T is calculated, whilst the M-step is utilised to re-estimate the parameter values from the E-step to achieve the best expected output.

C. The evaluation layer

The elements of the classification metrics are *CM={TP,TN,FP,FN}*, where *TP* (True positive) is the number of the correctly classified attacks, *TN* (True Negative) denotes the number of the correctly classified normal rows, *FP* (False Positive) is the number of the misclassified attacks, and *FN* (False Negative) refers to the number of the misclassified normal records. The accuracy is computed as the percentage of the correctly classified records over all the rows of data set, whether correctly or incorrectly classified [13], as reflected in the following Equation:

$$acc = (TP + TN)/(TP + TN + FP + FN) \quad (4)$$

The False Alarm Rate (FAR) is the rate of the misclassified to classified records, as denoted in Equation (6). Equations (5) and (6) allow calculation of the False Positive Rate (FPR) and the False Negative Rate (FNR), respectively.

$$FPR = FP/(FP + TN) \quad (5)$$

$$FNR = FN/(FN + TP) \quad (6)$$

$$FAR = (FPR + FNR)/2 \quad (7)$$

The Equations (4) to (7) evaluate the efficiency and reliability of Decision engine. It is acknowledged that the highest trusted detection is accomplished, when accuracy value closes to 100% and FAR closes to 0%.

VI. Results and Discussion

The second week of the KDD99 data set was replicated with the same UNSW-NB15 features to analyse the efficiency of these features. After that, the feature selection of ARM algorithm was developed using Visual studio C# 2008. we selected 100 rules to reduce the complexity of processing, and the number of required features is 11 to ensure at least 25% of all features. we computed the final results depending on the highest repeated features in three trails of implementations with different 100 rules. The proper features of the proposed ARM technique are applied to the UNSW-NB15 and KDD data sets, as presented in Table II and Table IV, respectively. Furthermore, the technique is applied to the original features of the KDD99, as shown in Table III.

TABLE II: UNSW-NB15 features of the proposed ARM

Category	Feature Numbers
Normal	11,34,19,20,21,37,6,10,11,36,47
DoS	6,11,15 16,36,37,39,40,42,44,45
Fuzzers	6,11,14,15,16,36,37,39,40,41,42
Backdoors	6,10,11,14,15,16,37,41,42,44,45
Exploits	10,41,42,6,37,46,11,19,36,5,45
Analysis	6,10,11,12,13,14,15,16,34,35,37
Generic	6,9,10,11,12,13,15,16,17,18,20
Reconnaissance	10,14,37,41,42,43,44,9,16,17,28
Shellcode	6,9,10,12,13,14,15,16,17,18,23
Worms	41,37,9,11,10,46,23,17,14,5,13

TABLE III: Original KDD99 features of the ARM

Category	Feature Numbers
Normal	3,37,36,30,38,12,22,39,16,20,15
DoS	30,4,39,25,26,38,5,32,2,3,37
Probes	36,23,24,12,6,39,8,10,16,7,11
R2L	1,10,7,6,21,38,9,8,3,27,31
U2R	13,17,14,10,15,14,30,33,31,36,3

The evaluation criteria of applying the Naïve Bayes (NB) and the EM clustering are computed in terms of accuracy (Acc.) and False Alarm Rate (FAR) to evaluate the complexity of these data sets. Table V compares the original features of the KDD99 and the generated features of the UNSW-NB15 on the kdd99 for each attack and normal records. Generally, the results show that the evaluation of the NB and EM on the UNSW-NB15 features is better than KDD99 features.

TABLE IV: KDD99 data set as UNSW-NB15 features of the ARM

Category	Feature Numbers
Normal	37,13,46,12,36,26,5,35,33,16,22
DoS	45,23,20,43,31,15,46,18,22,38,12
Probes	20,19,5,12,37,13,46,42,33,45,17
R2L	11,17,10,14,42,39,19,12,20,5,46
U2R	9,46,42,14,22,24,34,37,32,13,6

TABLE V: The evaluation of the original and UNSW-NB15 feature for the KDD99 data set

Category	The Original Features				The UNSW-NB15 Features			
	NB		EM		NB		EM	
	Acc.	FAR	Acc.	FAR	Acc.	FAR	Acc	FAR
Normal	96.7	3.22	93	6.79	70.4	29.42	93.7	12.21
R2L	46.01	51.7	32	77.3	83.1	16.8	32.6	73.5
DoS	76.7	23.3	66.7	35.9	95.9	4.06	95.3	4.8
Probes	69.8	30.1	60.8	39.11	86.2	13.4	59.1	40.6
U2R	20.89	79.1	10.2	89.61	54.7	47.56	13.7	86.8
Average	62.02	37.48	52.54	49.71	78.06	22.08	58.88	43.58

In Table VI, the evaluation criteria are applied on each record category of the UNSW-NB15 data set. The results show that these algorithms can not detect some record categories, such as the NB cannot identify Normal and Analysis records. While, the EM cannot detect Analysis, Backdoor and DoS records. Overall, the results of the other records are very poor.

TABLE VI: The evaluation of the UNSW-NB15 data set

Category	The UNSW-NB15 Features			
	NB		EM	
	Acc.	FAR	Acc.	FAR
Normal	0	100	45.27	54.73
Analysis	0	100	0	100
Backdoor	20	80	0	100
DoS	71.1	28	0	100
Exploits	54.6	46.41	74.5	24.67
Fuzzers	33.2	66.8	23.5	76.8
Generic	94.3	5.68	95.1	4.81
Reconnaissance	69.9	30	0	100
Shellcode	0	100	0	100
Worms	0	100	0	100
Average	37.5	62.58	23.83	75.80

The above results show that, the proposed feature selection ARM technique can be used to generate the best features of the KDD99 and UNSW-NB15. After comparing the UNSW-NB15 features with the KDD99 data set, the UNSW-NB15 features reflect the nature of both the contemporary normal and attack records. However, the algorithms of Decision engine cannot distinguish between the normal and attack rows, due to the similarity of their values relatively.

VII. Conclusions and Future work

In this paper, we compare the efficiency and reliability of the UNSW-NB15 and KDD99 features in terms of feature characteristics to distinguish between normal and abnormal records. The second week of the KDD99 data is replicated to the same features of the UNSW-NB15 data set. To remove irrelevant features for each record category, we developed a feature selection ARM technique. This technique applied to the KDD99 and the UNSW-NB15 data sets. This technique have some parameters, for instance the number of rules and the number of required feature, to reduce the computational time. The decision engine of the NIDS involves the Naïve Bayes and EM clustering models to evaluate the complexity of the two data sets in terms of accuracy and FAR. The experimental results show that, the evaluation criteria of the replicated UNSW-NB15 features of the KDD99 data set are better than the original KDD99 features. However, the evaluation criteria of the UNSW-NB15 data set show that, the existing algorithms of the decision engine can not detect several records categories, becuase of the similarities between the values of these records.

In future, we plan to develop a new NIDS algorithm to find out new patterns which are able to discriminate between the similar record values of each feature.

References

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

[8] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*. Springer, 2008, vol. 207.

[9] C. Zhang and S. Zhang, *Association rule mining: models and algorithms*. Springer-Verlag, 2002.

[10] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[2] A. Valdes and D. Anderson, "Statistical methods for computer usage anomaly detection using nides (next-generation intrusion detection expert system)," *Proceedings of the Third International Workship on Rough Sets and Soft Computing*, pp. 306–311, 1995.

[3] N. Moustaf and J. Slay, "Creating novel features to anomaly network detection using darpa-2009 data set," in *Proceedings of the 14th European Conference on Cyber Warfare and Security*. Academic Conferences Limited, 2015, p. 204.

[4] G. Vigna and R. A. Kemmerer, "Netstat: A network-based intrusion detection system," in *Journal of Computer Security*. Citeseer, 1999.

[5] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1, pp. 18–28, 2009.

[6] A. S. A. Aziz, A. T. Azar, A. E. Hassanien, and S. E.-O. Hanafy, "Continuous features discretization for anomaly intrusion detectors generation," in *Soft computing in industrial applications*. Springer, 2014, pp. 209–221.

[7] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 303–336, 2014.

[11] "Unsw-nb15," May 2015. [Online]. Available: http://www.cybersecurity.unsw.adfa.edu.au/ADFA%20NB15%20Datasets/

[12] N. Moustafa and J. Slay, "Unsw-nb15: A comprehensive data set for network intrusion detection," in *MilCIS-IEEE Stream, Military Communications and Information Systems Conference. Canberra, Australia*, 2015, in press.

[13] N. Moustaf and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, 2015, in press.

[14] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *Usenix Security*, 1998.

[15] J. Luo and S. M. Bridges, "Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection," *International Journal of Intelligent Systems*, vol. 15, no. 8, pp. 687–703, 2000.

[16] Z. Yanyan and Y. Yuan, "Study of database intrusion detection based on improved association rule algorithm," in *3rd IEEE International Conference, Computer Science and Information Technology*, vol. 4. IEEE, 2010, pp. 673–676.

[17] F. Shi, S. M. Bridges, and R. B. Vaughn, "The application of genetic algorithms for feature selection in intrusion detection," *Submitted for publication, GECCO*, 2000.

[18] M.-Y. Su, K.-C. Chang, H.-F. Wei, and C.-Y. Lin, "A real-time network intrusion detection system based on incremental mining approach," in *IEEE International Conference, Intelligence and Security Informatics*. IEEE, 2008, pp. 179–184.

[19] R. Amornchewin and W. Kreesuradej, "Probability-based incremental association rule discovery algorithm," in *International Symposium, Computer Science and its Applications*. IEEE, 2008, pp. 212–215.

[20] B. Nath, D. Bhattacharyya, and A. Ghosh, "Dimensionality reduction for association rule mining," *International Journal of Intelligent Information Processing*, vol. 2, no. 1, 2011.

[21] "Kddcup1999," April 2015. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[22] "Nslkdd," April 2015. [Online]. Available: https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/

[23] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM transactions on Information and system Security*, vol. 3, no. 4, pp. 262–294, 2000.

[24] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection," in *Recent Advances in Intrusion Detection*. Springer, 2003, pp. 220–237.

[25] A. Vasudevan, E. Harshini, and S. Selvakumar, "Ssenet-2011: a network intrusion detection system dataset and its comparison with kdd cup 99 dataset," in *Second Asian Himalayas International Conference, Internet (AH-ICI)*. IEEE, 2011, pp. 1–5.

[26] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications*, 2009.

[27] K. Hu, Y. Lu, L. Zhou, and C. Shi, "Integrating classification and association rule mining: A concept lattice framework," in *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*. Springer, 1999, pp. 443–447.

[28] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International Journal of Computer Science and Network Security*, vol. 7, no. 12, pp. 258–263, 2007.

[29] P. S. Bradley, U. Fayyad, and C. Reina, "Scaling em (expectation-maximization) clustering to large databases," Technical Report MSR-TR-98-35, Microsoft Research Redmond, Tech. Rep., 1998.