Using Feature selection for intrusion detection system

Ammar Alazab , Michael Hobbs, Jemal Abawajy, Moutaz Alazab
Deakin University, Australia
aalazab, mic, jemal.abawajy, malazab @deakin.edu.au

***Abstract*— A good intrusion system gives an accurate and efficient classification results. This ability is an essential functionality to build an intrusion detection system. In this paper, we focused on using various training functions with feature selection to achieve high accurate results. The data we used in our experiments are NSL-KDD. However, the training and testing time to build the model is very high. To address this, we proposed feature selection based on information gain, which can contribute to detect several attack types with high accurate result and low false rate. Moreover, we performed experiments to classify each of the five classes (normal, probe, denial of service (DoS), user to super-user (U2R), and remote to local (R2L). Our proposed outperform other state-of-art methods.**

Keywords- Feature selection; Intrusion detection; security; Anomaly base detection

I. INTRODUCTION

An intrusion can be defined as a series of actions aiming at compromising the security of a computer network system [1]. It attempts to bypass security mechanisms of computer systems that threaten the availability, integrity, or confidentiality of computer resources. Intrusions are had by malicious writer to gain information from the computer network. Intrusions can take several forms: external attacks, insider attacks [2].

Intrusion detection is a process of detecting and preventing intrusions activity. It is a significant way of defending the computer system from intrusions. However there are two kinds of intrusion detection: 1) host-based and 2) network-based. Each has a different away in detecting and securing data, and each of them has advantages and disadvantages [3]. Briefly, host-based IDSs examine internal data of computers systems, while network-based IDSs examine data exchanged between computers [4].

Two methods of data mining techniques have been applied in intrusion detection system. The first one is classification and the second one is clustering analyses. Description is more significant than prediction. However, if we were aiming to predict the activity of an intrusion based on behavior activities, prediction is more significant. Classification is learning a function for categorizing unseen data into one of several predefined classes based on a set of training, while clustering, the classes are not predefined at the stage of learning, and learning stage is to determine the classes in the database. If the purpose intrusion detection system is to distinguish an abnormal from a normal action, classification is more appropriate to accomplish this task. If the purpose of system seeks to identify the type of attack or malicious action, clustering is suit [6].

However, many researchers proposed to detect an attack in the KDD datasets [18][19]. Nevertheless, these proposals failed to achieve a good performance in terms of detection rate and false alarm rate using KDD dataset especially for new attacks.

Moreover, the exiting (IDS) examine all features to detect intrusion. However take all features may be misclassification of attack and take long time to build the model [18][23]. Thus, in this paper we propose a method of feature selection based on information gain to detecting both the old and the new attacks in KDD dataset. However, our proposed can successfully recognize the important features selection to building an (Intrusion Detection Systems) IDS.

After introducing the basic concepts of intrusion detection system in section I, the rest of the paper is organized as follows: Section II - The Related Work of Intrusion Detection; Section III – Presentation of Experimental Set up and Methodology; Section IV – Summary of Results

II. RELATED WORK

Most of research systems could effectively use training data to improve detection performance and minimize false alarm rates for known attacks. Researchers, however, missed many important new attacks, especially when the attack mechanism applied differed from the old attack. These results need further research in approach that could detect new attacks with high accuracy and low false alarm.

According to Shi-Jinn [7] studies revealed that not all researchers performed feature selection before classifier training. However, the feature selection is a significant part to identify the different types of attacks [8] [9] [10][22]. In their work [11], they examined the efficiency of two feature selection algorithms applying Bayesian networks (BN) and Classification and Regression Trees (CART) and an ensemble of BN and CART.

Literatures have shown some of features can be discarded, without affecting performance of the IDS [12]. Very little scientific efforts are diverted to model efficient IDS feature selection. IDS task is often modeled as a classification problem in a machine-learning context [13].

Nowadays, there are a few public datasets like KDD'99, DARPA 1998 and DARPA 1999, much related work considers these datasets for their experiments. Very few studies use non-public or their own datasets. This result shows that these public datasets are recognized as standard datasets in intrusion detection. Despite of many advances that have been achieved, existing IDSs still have some difficulties in improving their performance to meet the needs of detecting increasing types of attacks in high-speed networks. For example The SVM

978-1-4673-1157-1/12/$31.00 © 2012 IEEE

technique failed to manipulate a large dataset due to storage issues, or may take a long time to finish the training. This study used the KDD dataset to minimize the amount of time and improve the accuracy rate. Another difficulty was improvement of detection power for complex or new attacks without false alarms. As misuse IDSs used signatures of prior attacks, it has difficultly in detecting novel attacks.

The main advantage of anomaly detection systems is it can detect new types of attacks by built models of normal behaviors; the false alarm rates in anomaly-based IDSs are usually high.

III. EXPERIMENTAL SETUP & METHODOLOGY

This section describes our methodology, intrusion detection dataset, feature selection, training and testing dataset before and after feature selection approach and experimental setup.

Our experiment aims to learn from NLS-KDD dataset while performing Prediction process whether a packet should be normal or abnormal.

A. METHODOLOGY

Our purpose is to design a classifier with feature selection, which could give the best accuracy for each category of attack patterns. The first step is to carefully construct the different connectional models to achieve the best generalization performance for classifiers. We performed experiments on Intel Processor: 3.64 GHz, Memory (RAM): 96 GB RAM with Windows XP operating system. Also we performed 5-class classification of dataset using WEKA machine learning tool to classify KDD dataset. The stages of experiment are described as follows and shown in figure 1.

1) Feature selection stage: In this stage, an information theoretic feature selection approach [14]is applied to normalized Training and Test Dataset for generating reduced feature set selection ..

2) Classification stage: classification stage involves two phases namely training phase and testing phase.

3) Analysis of the result: After testing phase, we compute the accuracy rate, false alarm rate and the time to build the model.

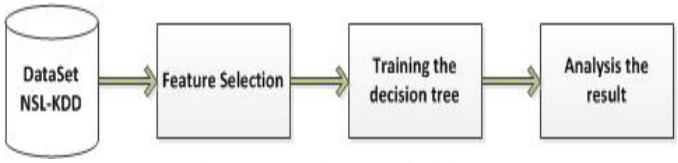

Figure 1. Experiment methodology.

B. INTRUSION DETECTION DATASET:

The NSL-KDD intrusion detection benchmark consists of categories of attacks representing generalizations of specific attack types (Tavallaee et al. 2009). These main categories represent classifications of types of behavior that can be grouped logically together. So, for each category, there are multiple attack types. Table shows these attack categories along with the 22 attack types. The KDD learning set distribution is shown in figure 2.

TABLE I. ATTACK TYPES

Attack type	Attack pattern
Probe	Ipsweep, nmap, portsweep, satan, mscan, saint
Dos	back, land, neptune, pod, smurf, teardrop, apache2, mailbomb, processtable, udpstorm
U2R	Buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, xlook, xsnoop, snmpguess, worm

In addition, each attack category (U2R, R2L, DoS, or Probing) is composed of a given number of attacks. Some of these attacks are known (For, U2R is composed of eight attacks buffer overflow, httptunnel, loadmodule,), among them we can distinguish 4 new attacks: httptunnel, ps, sqlattack, and xterm (see Table 1). We will focus in this study on applying our original approach to detect both the known and the new attacks of the U2R category.

In the KDD99 data set, each data record corresponds to the features of a connection in the network data flow. Each connection is labeled either as normal or as an attack, with exactly one specific attack type. The data records are all labeled with one of the following five types [21]:

- Denials-of Service (DoS): the attacker try to make the computer service unavailable. An attacker send many packet to the victim such as (. apache, smurf, Neptune, Ping of death, back, mail bomb, udpstorm, SYNflood, etc.).Probing or surveillance attacks have the goal of gaining knowledge of the existence or configuration of a computer system or network. Port Scans or sweeping of a given IP-address range typically fall in this category
- Probe Layer attacks have the goal to gain user information from a network such as (mscan ,saint, port sweep, nmap)
- User-to-Root (U2R) attacker access to a user data on the computer system.
- Remote-to-Local(R2L) attackers send malicious string to a victim over the network, the attacker to expose the client vulnerabilities and exploit privileges which a local user would have on the computer (e.g. xclock, dictionary, guest password, phf, send mail, xsnoop, etc.).

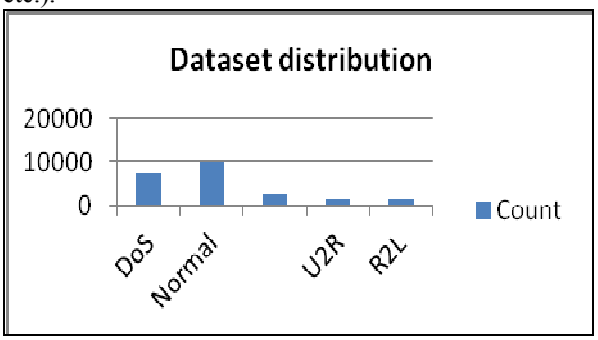

FIGURE 2. DATASET DISTRIBUTION

TABLE II. KDD DATASET FEATURES

Label	Network data feature	Label	Network data feature	Label	Network data feature	Label	Network data feature
A	duration	L	Logged in	W	count	AH	dst_host_same_srv_rate
B	protocol-type	M	num_comprised	X	srv_count	AI	dst_host_diff_srv_rate
C	service	N	root_shell	Y	serror_rate	AJ	dst_host_same_src_port_rate
D	flag	O	Stu attempted	Z	srv_serror_rate	AK	dst_host_srv_diff_host_rate
E	src_bytes	P	num_root	AA	rerror_rate	AL	dst_host_serror_rate
F	dst_bytes	Q	Num of file	AB	srv_rerror_rate	AM	dst_host_srv_serror_rate
G	land	R	Number of shell	AC	same_srv_rate	AN	dst_host_rerror_rate
H	wrong_fragment	S	num_access_files	AD	diff_srv_rate	AO	dst_host_srv_rerror_rate
I	urgent	T	num_outbound_cmds	AE	srv_diff_host_rate		
J	hot	U	Is host login	AF	dst_host_count		
K	num_failed_logins	V	Is guest login	AG	dst_host_srv_count		

C. FEATURE SELECTION

The four cited attack categories contain 22 training attack types. KDD dataset–contains 41 features as shown in table 2. Twenty one of these features describe the connection itself and 19 of them describe the properties of connections to the same host in last two seconds. These features can be categorized as either Content Features of Network Connection Records (see table 3), Connection based features (see table 4) or Traffic feature (see table 5). Table 2 illustrates all the KDD features.

TABLE III. CONTENT FEATURES OF NETWORK CONNECTION RECORDS

feature	description	value type
hot	number of "hot indicators"	continuous
failed logins	number of failed login attempts	continuous
logged in	1 - successfully logged in; 0 - otherwise	discrete
compromised	number of "compromised" conditions	continuous
root shell	1 - root shell is obtained; 0 - otherwise	discrete
su	1 - "su root" command attempted; 0 - otherwise	discrete
file creations	number file creation operations	continuous
shells	number of shell prompts	continuous
access files	number of write, delete, and create operations on access control files	continuous
outbound cmds	number of outbound commands in a ftp session	continuous
hot login	1 - the login belongs to the "hot" list (e.g., root, adm, etc.); 0 - otherwise	discrete
guest login	1 - The login is a "guest" login (e.g., guest, anonymous, etc.) ; 0 - otherwise	discrete

TABLE IV CONTENT FEATURES OF NETWORK CONNECTION RECORDS

feature	description	value type
duration	length (number of seconds) of the connection	continuous
protocol type	Type of the protocol like FDDI. tcp, udp, etc.	discrete
service	is a service hosted on a computer network like destination Domain Name System (DNS), DHCP, NetBIOS	discrete
source bytes	number of data bytes from source to destination	continuous
dst bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 - connection is from/to the same host/port; 0 - otherwise	discrete
wrong fragment	number of "wrong" fragments	continuous
urgent	number of urgent packets	continuous

TABLE V: TRAFFIC FEATURE

feature	description	value type
count	number of connections to the same host as the current connection in the past 2 seconds	continuous
serror %	% of connections that have "SYN" errors	continuous
rerror %	% of connections that have "REJ" errors	continuous
same srv	% of connections to the same service	continuous
diff srv %	% of connections to different services	continuous
srv count	number of connections to the same service as the current connection in the past 2 seconds	continuous
Null	the following features refer to these same-service connections	
srv serror %	% of connections that have "SYN" error	continuous
srv rerror %	% of connections that have "REJ" errors	continuous
srv diff host %	% of connections to different hosts	continuous

TABLE VI KINDS OF FEATURES

(KDDCup 99) features	Meaning
content-based features	number of packets, acknowledgments, data bytes from src to dest)
time-based traffic features	included number of connections or different s from the same source or the same destination considering recent time interval
connection based features	included number of connections from same source or to same destination or with the same service considering in last N connections

Due to some features made redundant, increased computation time on extra features and impact on accuracy of IDS. We improved the performance of classification algorithm by using useful features. Feature selection is an important for intrusion in order to increase the accuracy and the performance. We present our work for identifying intrusion by using features selection and evaluating the applicability of these features in detecting intrusions. We also present different feature selection methods for intrusion detection and rank the importance of the

features by the absolute values of weights according to the ranking method.

Each feature will be rated as "very important", "important", or "unimportant" according to the following rules:

1. ***If*** *accuracy high* ***and*** *training time low,* ***then*** *the feature is important*
2. ***If*** *accuracy* ***high*** *and training time low,* ***then the*** *feature is very important*
3. If accuracy low and training time high, then ***the feature is unimportant***
4. ***If*** *accuracy low* ***and*** *training time high,* ***then*** *the feature is unimportant*
5. ***If*** *accuracy unchange* ***and*** *false alaram decreased ,* ***then*** *the feature important*

According to these rules, the 41 features are rated into the three kinds
$\{< Very\ important\ features >, < Important\ features > , (Unimportant\ features)\}$

For all of the five classes of attack, we applied information gain into 41 features of KDD dataset. Then we considered the values of features are $\{< Very\ important\ features, < Important\ features >\}$

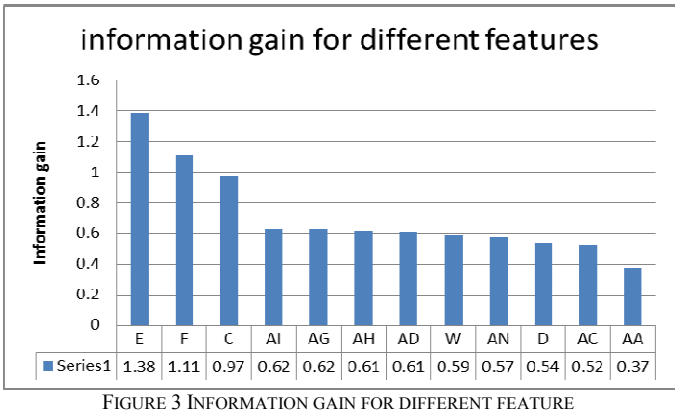

FIGURE 3 INFORMATION GAIN FOR DIFFERENT FEATURE

We applied these criteria of selection using information gain as shown in figure 3. Information gain, originally used to compute splitting criteria for decision trees, is often used to find out how well each single feature separates the given data set. The overall entropy I of a given dataset S is defined as (Gray 2010)

$$I(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

Where C denotes the total number of classes and pi the portion of instances that belong to class i. The reduction in entropy or the information gain is computed for each attribute according to $IG(S,A) = I(S) - \sum_{v \in A} \frac{|S_{A,v}|}{|S|} I(S_v)$ where va value of is A and $S_{A,v}$ the set of instances where Avalue has v. We applied information gain ~~on to~~ into 41 features of KDD dataset as the quality of the feature selection is one of the most important factors that affect the effectiveness of IDS.

IV. EMPIRICAL STUDY

In our experiments, we have evaluated various algorithms based on the following standard performance measures

- True positive (TP): Number of correctly identified malicious code.
- True negative (TN): Number of correctly identified benign code.
- False positive (FP): Number of wrongly identified benign code, when a detector identifies benign file as a malware.
- False negative (FN): Number of wrongly identified malicious code, when a detector fails to detect the malware because the virus is new and no signature is yet available.
- Detection Rate (DR) :the number of correctly classified positive examples divided by the total number of examples that are classified as positive , according the following equation

$$DR = \frac{TP}{TP + FP}$$

- Precision of a classifier is the proportion of positive predictions made by the classifier that are true.
- Recall is the percentage of correct positive that are truly detected by the classifier.
- F-measure is defined as the harmonic mean of recall and precision according the following equation

$$\text{F-measure} = \frac{2 * \text{Re}call * \text{Pr}ecision}{\text{Re}call + \text{Pr}ecision}$$

TABLE VII CONFUSION MATRIX

	Predicted attack	Predicated normal
Actual attack	True positive (TP)	False negative (FN)
Actual normal	False positive (FP)	True negative (TN)

TABLE VIII CONFUSION MATRIX OF IDSS

	Normal	Probe	U2R	R2L	DoS
Normal	X11	X12	X13	X14	X15
Probe	X21	X22	X23	X24	X25
U2R	X31	X32	X33	X34	X35
R2L	X41	X42	X43	X44	X45
DoS	X51	X52	X53	X54	X55

In general the confusion matrix is shown in table 7. In table 8 shown confusion matrix, the element Xij $(1 \leq i \leq 5, 1 \leq j \leq 5)$ denotes the number of records that belong to class i and were classified as class j by IDSs. Therefore, based on the

confusion matrix, we can easily compute other performance criteria such as the detection rate of class i:

$DR\ (i) = \frac{Xii}{\sum_{i=1}^{5} Xij}$ And the false alarm rate of IDSs can be computed by $F\ (i) = 1 - \frac{Xii}{\sum_{i=1}^{5} Xij}$

A. EXPERIMENT USING DECISION TREES FOR SUPERVISED LEARNING

Decision tree algorithms are supervised learning algorithms which recursively partition the data base on its attributes, until some stopping condition is reached. This recursive partitioning gives rise to a tree-like structure. The aim is that the final partitions (leaves of the tree) are homogeneous with respect to the classes, and the internal nodes of the tree are decisions about the attributes that were used to reach the leaves. The decisions are usually simple attribute tests, using one attribute at a time to discriminate the data. New data can be classified by following the conditions defined at the nodes down. J.R. Quinlan has popularized the decision tree approach. The latest public domain implementation of Quinlan's model is C4.5 [16]. The Weka classifier package has its own version of C4.5 known as J48. WEKA is open source Java code created by researchers at the University of Waikato in New Zealand [17].

We used 10-fold Cross-Validation. The data is divided randomly into 10 parts in which the class is represented in approximately the same proportions as in the full dataset. Each part is held out in turn and the learning scheme trained on the remaining nine-tenths; then its error rate is calculated on the holdout set. The learning procedure is executed a total of 10 times on different training sets, and finally the 10 error rates are averaged to yield an overall error estimate.

B. RESULTS

An initial analysis was performed to determine the accurate result and false-alarm rates. In terms of training and testing speed, it is obvious in figure 4 that the feature selection and tree decision is five times faster in terms of the time to build the model without feature selection.

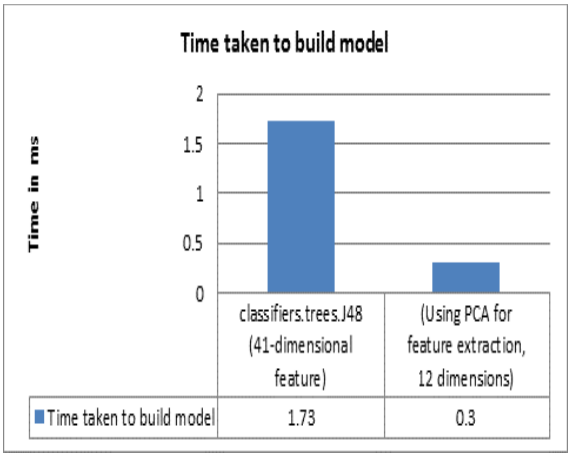

FIGURE 4 PROCESSING SPEED COMPARISON FOR J4TREE.J48 FEATURE-41-FEATURE VS J4TREE.J48 FEATURE-21- FEATURE

The detailed analysis of accuracy by j48 classification using 12 features is shown in table IX, and table X shows the detailed accuracy by j48 classification using 41 features.

TABLE IX. DETAILS ACCURACY BY J48 CLASSIFICATION USING 12 FEATURES

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.997	0.002	0.997	0.997	0.997	0.999	DoS
0.985	0.014	0.982	0.985	0.983	0.995	normal
0.978	0.002	0.986	0.978	0.982	0.994	Probing
0.972	0.002	0.972	0.972	0.972	0.992	U2R
0.913	0.006	0.919	0.913	0.916	0.989	R2L
0.982	0.007	0.982	0.982	0.982	0.996	Weighted Avg.

TABLE X DETAILS ACCURACY BY J48 CLASSIFICATION USING 41 FEATURES

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.996	0.002	0.996	0.996	0.996	0.998	DoS
0.985	0.014	0.982	0.985	0.983	0.995	normal
0.98	0.001	0.989	0.98	0.984	0.995	Probing
0.973	0.002	0.971	0.973	0.972	0.991	U2R
0.919	0.005	0.925	0.919	0.922	0.99	R2L
0.983	0.007	0.983	0.983	0.983	0.995	Weighted Avg.

Table XI shows the confusion matrix tree j48 classification by using 12 features, 9562 of the actual 'normal' test set were detected to be normal. For Precision, 0.982% was detected correctly. In the same way, for 'DoS' 7432 of the actual 'attack' test set were correctly detected; for 'Probe' 2367 of the actual 'attack' test set were correctly detected; for 'U2R' 1391 of the actual 'attack' test set were correctly detected; for 'R2L' 1391 of the actual 'attack' test set were correctly detected;

TABLE XI CONFUSION MATRIX USING TREE J48 CLASSIFICATION BY USING 12 FEATURES

	DoS	Normal	Probe	U2R	R2L
DoS	7432	15	8	3	0
Normal	11	9562	15	20	103
Probe	12	35	2367	4	3
U2R	0	20	4	1391	16
R2L	3	109	7	13	1391

Many intrusion detection systems are proposed by researchers to achieve a high accurate result for the classification. As shown in Table XII, the proposed classifiers demonstrate better performance in classification rate category among mentioned models.

TABLE XII PERFORMANCE COMPARISON

Model	Classification rate					DR	False Alarm
	Normal	Probe	R2L	DoS	U2R		
Proposed model	98.2	99.6	99.7	97.2	92.5	98.2	0.17
ARTMAP [2]	99.82	84.93	99.72	17.52	59.28	96.81	0.18
PNrule [1]	99.5	73.2	96.9	6.6	10.7	91.1	0.4
Winner of KDD in 2000 [18]	99.5	83.3	97.1	13.2	8.4	91.8	0.6
Runner up of KDD in 2000 [19]	99.4	84.5	97.5	11.8	7.3	91.5	0.6
ESC-IDS [20]	98.2	84.1	99.5	14.1	31.5	95.3	1.9
MLP with 38 selected features [15]	99.6	75.5	99.7	14.3	32.7	94.9	0.36

V. CONCLUSION

Our research illustrated the significance of using a suitable feature with appropriate classification for modeling IDSs. We evaluated the effective value of the decision tree as the data mining method for the IDSs with appropriate features. We concluded before training, that the step of feature selection may be considered. The process of feature selection identifies which features are more useful than the others. This has the benefit of generally improving system performance by eliminating irrelevant and redundant features. Therefore, feature selection could improve some certain level of classification accuracy in intrusion detection.

Our proposed method is evaluated on a huge data set. The features selection in this paper were successfully achieved high result in term the accuracy , false alarm and the time to build the module using appropriate feature based on information gain.

VI. REFERENCE

[1] R. Agarwal and M. V. Joshiy, "PNrule: A new framework for learning classi er models in data mining (a case-study in network intrusion detection)," Citeseer2000.

[2] M. Sheikhan, et al., "Application of Fuzzy Association Rules-Based Feature Selection and Fuzzy ARTMAP to Intrusion Detection," Majlesi Journal of Electrical Engineering, vol. 5, 2011.

[3] R. Beghdad, "Efficient deterministic method for detecting new U2R attacks," Computer Communications, vol. 32, pp. 1104-1110, 2009.

[4] R. Chattemvelli and R. Sridevi, "GA Approach for Network Intrusion Detection," International Journal of Research and Reviews in Information Sciences (IJRRIS), vol. 1, 2012.

[5] M. Kantardzic, Data mining: concepts, models, methods, and algorithms: Wiley-IEEE Press, 2011.

[6] L. Han, "Using a Dynamic K-means Algorithm to Detect Anomaly Activities," 2011, pp. 1049-1052.

[7] C. F. Tsai, et al., "Intrusion detection by machine learning: A review," Expert Systems with Applications, vol. 36, pp. 11994-12000, 2009.

[8] S. Huda, et al., "Exploring novel features and decision rules to identify cardiovascular autonomic neuropathy using a hybrid of wrapper-filter based feature selection," 2010, pp. 297-302.

[9] V. Bolón-Canedo, et al., "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset," Expert Systems with Applications, vol. 38, pp. 5947-5957, 2011.

[10] F. Amiri, et al., "Improved feature selection for intrusion detection system," Journal of Network and Computer Applications, 2011.

[11] S. Chebrolu, et al., "Feature deduction and ensemble design of intrusion detection systems," Computers & Security, vol. 24, pp. 295-307, 2005.

[12] H. F. Eid, et al., "Bi-Layer Behavioral-Based Feature Selection Approach for Network Intrusion Classification," Security Technology, pp. 195-203, 2011.

[13] W. Fan, et al., "Unsupervised Anomaly Intrusion Detection via Localized Bayesian Feature Selection," 2011, pp. 1032-1037.

[14] R. M. Gray, Entropy and information theory: Springer Verlag, 2010.

[15] M. Sheikhan, et al., "Effects of feature reduction on the performance of attack recognition by static and dynamic neural networks," World Applied Sciences Journal, vol. 8, pp. 302-308, 2010.

[16] J. R. Quinlan, C4. 5: programs for machine learning: Morgan kaufmann, 1993.

[17] M. Hall, et al., "The WEKA data mining software: an update," ACM SIGKDD Explorations Newsletter, vol. 11, pp. 10-18, 2009.

[18] R. Kohavi, et al., "KDD-Cup 2000 organizers' report: peeling the onion," ACM SIGKDD Explorations Newsletter, vol. 2, pp. 86-93, 2000.

[19] I. Levin, "KDD-99 Classifier Learning Contest: LLSoft's Results Overview," SIGKDD explorations, vol. 1, pp. 67-75, 2000.

[20] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," Computer Communications, vol. 30, pp. 2201-2212, 2007.

[21] Kohavi, R., Brodley, C. E., Frasca, B., Mason, L., & Zheng, Z. (2000). KDD-Cup 2000 organizers' report: peeling the onion. *ACM SIGKDD Explorations Newsletter, 2*(2), 86-93.

[22] Alazab, M, Venkataraman, S & Watters, P 2010, 'Towards Understanding Malware Behaviour by the Extraction of API Calls', in pp. 52-9.

[23] Alazab, M, Venkatraman, S & Watters, P 2011, 'Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures', in pp. 171-81..