NETWORK INTRUSIONS CLASSIFICATION USING DATA MINING APPROACHES

by Slamet A.

Submission date: 17-Mar-2023 11:16AM (UTC+0700)
Submission ID: 2039102532
File name: 02._17Vol99No7.pdf (487.73K)
Word count: 7989
Character count: 42641

NETWORK INTRUSIONS CLASSIFICATION USING DATA MINING APPROACHES

SLAMET[1,2*], IZZELDIN IBRAHIM MOHAMED[2]
[1]Department of Information System, Faculty of Technology and Informatics,
Universitas Dinamika,
Jl. Kedung Baruk 98 Surabaya, Jawa Timur 60298, Indonesia
[2] Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang,
26600 Pekan, Pahang, Malaysia

E-mail: [1,2*]slamet@dinamika.ac.id, [2]izzeldin@ump.edu.my

ABSTRACT

Intrusion Detection System has an important task in detecting threats or attacks in the computer networks. Intrusion Detection System (IDS) is a network protection device used to identify and check data packets in network traffic. Snort is free software used to detect attacks and protect computer networks. Snort can only detect misuse attacks, whereas to detect anomaly attacks using Bayes Network, Naive Bayes, Random Tree, LMT and J-48 Classification Method. In this paper, the experimental study uses the KDDCUP 99 dataset and the dataset taken from Campus Network. The main objective of this research is to detect deceptive packets that pass computer network traffic. The steps taken in this study are data preparation, data cleaning, dataset classification, feature extraction, rules snort for detecting, and detecting packet as an attack or normal. The result of the proposed system is an accurate detection rate.

Keywords: *Classification, Intrusion, Snort, Anomaly, Misuse*

1. INTRODUCTION

All types of attacks on a computer network must be resisted so that the network looks reliable. As we know that computer networks and services have become the backbone for activities in industry, business, and all aspects of human life. An intruder attack in the computer network becomes an enemy of security personnel and all those who have the responsibility to provide protection for the network and its users. Instructive information security with information storage capabilities where coordinated warehouse information checks are used as a basic principle of work [1].

User accounts, personal files, passwords and all network resources must be protected by network administrators and security personnel to create a comfortable and secure network environment. Various methods are used by intruders to carry out their attacks on the network. Denial of service (DoS) is one of the most commonly used ways to attack network resources and make network services paralyzed. Different types of DoS attacks have their own behavior in attacking network resources to achieve goals, but the point is that the network is not available to its users [2]. Remote to user (R2L) is a type of computer network attack, where sets of packets are sent to other computers by intruders who actually do not have permission to access as users. User to root (U2R) attacks are another type of attack where network resources are tried with various attempts to be fully accessed by intruders like the original user [3]. Probing is the next type of attack where network devices are scanned by intruders to find weaknesses in topology or open ports. This step is used by intruders to make illegal access to personal information at the time they will specify. Examples of this attack through the network, such as ipsweep, nmap, and portsweep.

Intrusion detection systems (IDS) become an important part in capturing attacks like this because IDS is able to work against all attacks. Classification techniques are used by IDS to detect and decide whether each packet that passes through the network is a normal packet or attack packet (eg DOS, U2R, R2L, PROBE).

All types of intruder attacks, such as DOS, R2L, U2R, and PROBE use the repository dataset from Knowledge Discovery in Databases (KDD). The KDD dataset is used to evaluate classifiers. The methodology used is first, to do the preprocessing step in the KDD dataset. The dataset is used in predetermined environments. Then the classifier is examined and evaluated to determine which is more accurate than the others in detecting all the attacks being studied (DOS, R2L, U2R, and PROBE).

On the other hand, Snort is software that can also detect network intrusion. Snort is able to analyze, search and match protocol content in the network. Snort is capable of detecting various attempted attacks such as port scanning, OS fingerprint attempts, CGI attacks, buffer overflow and Samba probes. ease of configuration and flexibility in entering rules into the database are examples of the benefits of snort. Multi-variant packets are investigated and detected by snort using Sniffer_mode, Network_Intrusion_Detection_System_mode and Packet_Logger_mode. Snort architecture is shown in figure 1.

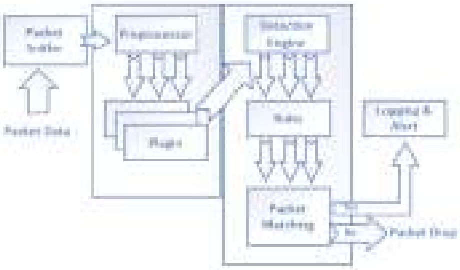

Figure 1: Snort Architecture

Many researchers have used data mining to improve IDS by offering external detection that identifies the presence of boundaries in normal network activity. Normal and abnormal activities can be distinguished in this way [4]. Data Mining can also be applied in identifying the presence of attacks in the system and increasing the level of detection accuracy. In addition, classification and clustering models can be developed to distinguish between packets with normal and abnormal behavior. Procedures for detecting intrusion of complete and accurate network traffic are classified as a classification problem.

The classification model can identify dangerous attacks, reduce false alarms and improve the accuracy of detection rates. In produce efficient IDS, several data mining algorithms have been proposed previously such as AdaBoost [5], KNN [6], Support Vector Machines [7] etc. All of these algorithms are integrated with different models in distinguishing dangerous attacks and normal behavior to detect unknown attacks. The classification of IDS depends on their detection techniques, architecture and post-detection activities [11].

Due to the increasing diversity of attacks and new attacks that are developed every day it must provide the system with flexibility and adaptability. Anomaly-based network intrusion detection can detect new attacks before security analysts even learn about it, while rules/ signature-based network intrusion detection systems are good at detecting known and defined attacks. The two approaches together create a system for identifying disruptive behavior. Rule-based uses a signature based approach and machine learning based uses anomaly based approach. Rule-based learning is utilized to classify packets as normal or abnormal, and to reduce false positive and false negative rates in intrusion detection, machine learning based system is used [4]. Rule-based can identify known attacks. Meanwhile, machine learning can learn without being programmed directly, starting by providing input of some knowledge, so that it can analyze, interpret, and test the knowledge gathered.

Therefore, this approach provides assurance that no packet with intrusion can pass through our network system. In this paper, we introduced rule-based learning system and machine learning classification. Rule-based is used to capture real-time data by using the SNORT and machine learning based is used to solve the classification problem. In this study, researchers have worked to improve the Bayes (Bayes Network, Naive Bayes) and Decision Tree (Random Tree, LMT, J48) as machine learning based. These Algorithms as an effort to deal with ever-increasing and changing intruder attacks. It also protects the network against internal and external attacks by increasing the accuracy of the detection rate and performance of anomalous IDS. We used KDD dataset to build machine learning model [8].

According to our research hypothesis, In order to make attacks identified by SNORT more verified in real time, trained machine learning models can be applied in SNORT classification data. Our approach can be divided into two parts; In the first, SNORT is configured according to some predefined rules to capture network traffic in real time and this information is stored in the database. In the second part, a machine learning model is trained using KDD dataset. Machine learning (Data mining classification) based is used to solve the classification problem and bring up the detection rate accuracy.

The contributions of this paper are as follows. (1) A Real-time internet traffic dataset has been developed using snort and then a reduced feature dataset has also been developed using an attribute selection algorithm. (2). Full feature and reduced feature datasets, Bayes and Decision Tree algorithms have been used for intrusion traffic classification: Bayes Network, Naive Bayes, Random Tree, LMT, and J48 Algorithms. The performance of all these classifiers is analyzed on the basis of the classification accuracy of classifiers.

Related Works

IDS detects attacks in the network to protect the system from intruders. Based on the attack detection method, IDS is divided into two main classifications. The first is misuse detection and the second is anomaly detection. Anomaly detection method can be used to detect strange behavior of users in network traffic.

Comparative results between the SSENet-2011 dataset and snort performance in real-time data have proven that the scope of snort work is limited and Vasudevan, AR et.al [8] uses sophisticated methodologies to improve snort performance. Claude Turner, et.al [9] has investigated and proposed Snort-based Detection. Tests are carried out in the network traffic using five different versions of the snort rules. The result is that around 88% of the configured rules fail to provide protection related to network security. There is plenty of room for writing more complex snort rules to improve security standards. Anna L. Buczak [10] uses machine learning techniques and data mining approaches to detect abuse and anomaly attacks in cyber traffic in real time.

Innovative Hybrid detection has been done by Syed Rizvi, et.al [11], they developed a honeypot-based scheme in a virtual environment. The result is an inefficient rate of reduction of a hybrid signature based snort system. The EC logic encoding technique has been proposed by Mohsen Rouachedah, Hassen Sallay [12], but has limitations in dealing with various threats. To detect incorrectly injected data, Bo Sun et.al [13] uses the "Extended Kalman Filter" approach. The result is better accuracy in WSNs.

The security of the cellular agent itself is an obstacle to intrusion detection. Intrusion detection using cellular agents developed by Saidi [14] can capture flooding attacks in the cloud environment such as DDoS and DoS attacks.

The selected attribute feature was extracted by Avrim L. Bluma and Pat Langley [15] using a machine learning algorithm. Web content and a large amount of low quality information have been used as input data for intrusion detection.

Artificial Neural Networks (ANN) and Fuzzy clustering (FC) as building blocks of IDS have been proposed to find problems and attacks on the network. However, there are limitations such as having a lack of accuracy in a barrage of attacks. The researchers took over the boundaries by dividing the heterogeneous training set into a homogeneous training set of parts, by reducing the complexity of each training group, so that detection performance improved and system backups could be retrieved successfully by using restore points [16].

The classifier selection model conducted by HuyAnh Nguyen and Deokjai Choi [17] extracted 49,596 KDD dataset samples and compared a set of classifiers under the control environment. A different approach to dealing with KDD datasets is carried out by Kamlesh Lahreet et al [18], matlab is used to simulate supervised and unsupervised methods, and researchers test the technique with fuzzy rules to identify system performance. L. Breiman [19] focused on random forests and how to combine them among tree predictors, errors in random forests as a limit on the number of trees in the forest have been proposed by researchers.

In the current scenario, the intrusion traffic classification uses machine learning techniques [5], [6], [7] which is based on network training using a set of previous examples. Then this trained network is used to predict the unknown class of test samples.

In this paper, a real-time internet traffic dataset using SNORT has been developed, also using an attribute selection algorithm and the development of a reduced feature dataset. This full feature and reduced feature datasets have been used five ML algorithms for intrusions traffic classification: Bayes Network, Naive Bayes, Random Tree, LMT, and J48 Algorithms. Classification accuracy and classifier precision values are used as performance variables of these classifiers.

2. METHODOLOGY

2.1. Architecture

The proposed intrusion detection system is divided into seven modules and the activities of each phase are shown in figure 2.

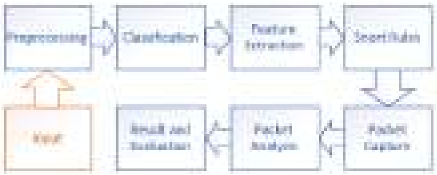

Figure 2: Proposed Architecture of an IDS

2.2. Preprocessing

Preprocessing is used to remove inconsistent data. In addition, preprocessing can handle missing values, eliminate noise from data sets and eliminate frequency attributes from unnecessary datasets. WEKA tools are used for preprocessing [14, 20, 21]. The results are four main categories of groups with 21 types of attacks in a number of different events. The number of instances for each type of attack can be seen in Figure 1. The experiment was carried out by randomizing all instances after extracting the KDD dataset complete with 148,753 instances, as shown in table 1.

Table 1: Number of instances for each type of attack

Type of Attack	Instances Number	
	Before Organization	After Organization
Back	2.205	70
Buffer Overflow	30	10
FTP Write	8	2
Guess Password	53	10
Imap	12	4
IP Sweep	12.481	382
Load Module	9	2
MultiHop	7	2
Neptune	1.072.017	32.827
Nmap	2.316	70
Perl	3	1
Phf	4	1
POD	264	10
Portsweep	10.413	318
Rootkit	10	5
Satan	15.892	487
Smurf	2.807.866	85.983
Spy	2	1
Teardrop	979	30
Warez Client	1.020	31
Warez Master	20	7
Normal	972.781	28.500

2.3. Classification

After the KDD preprocessing stage, the next step is the classification experiment using Bayesian (Bayes network, Naive Bayes) and tree algorithm classifier (J-48, Random Tree, LMT). Researchers use Weka software as a classification tool. The output of this phase is to produce a file for Anomaly Classification and a file for Misuse Classification provided by the kddcup99 dataset [22, 23]. Both of these files are used to predict the attack.

2.3.1 Bayes Network

This is a grouping for guided learning using independent feature assumptions. This learning theory represents the distribution of the naive Bayesian classifier. Bayes Network uses a variety of different search algorithms and quality measurement methods which Bayes Network is an enhancement to Naïve Bayes [24]. Bayesian networks can help understand the world being modeled. BayesNet can be the best because it can model mysterious facts in the web of decisions. Bayes Net is able to make smart, justifiable, and measurable decisions to improve classification performance. BayesNet helps for diagnosis, prediction, modeling, monitoring and classification [25].

The Bayesian Classifier consists of two phases: first, if the agent has an idea and knows the class, then he can predict the value of other features. Second, if the agent has no idea or does not know the class, then the Bayes rule can be used to predict the given class.

The theory of the Bayesian Network is shown in Equation (1), where the symbol *D* indicates the training data, the probability of hypothesis h.

$$P(h|D) = \frac{p(D|h)p(h)}{p(D)} \qquad (1)$$

The symbol is to :
P(h|D) : probability of posterior
P(D|h) : probability of condition
P(h) : previous probability of h
P(D) : marginal probability of D

2.3.2 Naive Bayes

This type of classifier is a simple probabilistic classifier where in defining discriminatory learning, new class values are predicted by returning p (y|x), and calculating probabilistic for each class in the dataset. Further main formulations for Nave Bayes can be read at [27].

The dataset attribute [illegible]∈X is linked by the Naive Classification as input to the class label Z∈ {1,2, ,C}, where Z is the class room and X is the attribute space. For example X = IRD, then D is a real number. The Naive classification model that uses continuous and discrete attributes is called the multi-label problem model. Discriminates model is a learning function that directly calculates class $p = \left(\frac{y}{x}\right)$. The main objective is to study conditional classes in non-linear and multi-label problems. For this purpose, equation (2) is used:

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p\left(\frac{x}{y}\right)p(y)}{\sum_{y'=1}^{c}\frac{p\left(\frac{x}{y'}\right)}{p(y')}} \qquad (2)$$

The output is generated by the Naïve Classifier based on the max argument function, as seen in equation (3):

$$f(x) = y'(x) = argmax\ \{\left(p\left(\frac{x}{y}\right)\right)\} \qquad (3)$$

Probability classifiers have advantages in terms of [26]:

a. The choice to reject or ignore the predicted results when not sure of the results due to human efforts.
b. Allow to change the learning function and combine the probability function to achieve the highest performance. If the direct learning function $p = \left(\frac{y}{x}\right)$) is used and the probability function is changed; then $p = \left(\frac{y}{x}\right)$ does not need to be recalculated.
c. A balanced class of several data sets that has an unbalanced class. This means that if one million normal network traffic records are owned and there is only 1 abnormal out of 1000 records, the training dataset can be trained and easily reach 99% accuracy by using class always = normal.

In solving that problem-balanced classes can use equation(4) and equation (5).

$$Pbal(y|x) \propto= P(x,y)Pbal(y) \qquad (4)$$

$$Ptrue(y|x) \propto Ptrue(x,y)\ Ptrue(y) \propto \frac{p\left(\frac{y}{x}\right)}{pbal(y)}\ Ptrue(y) \qquad (5)$$

Data sets that contain a mixture of feature types, such as data sets where each feature vector represents different data types (text, images, numbers) are very useful as a combination of models. Two or more classifiers can be constructed by two or more types of attributes that use a combination of models, such as $p\left(\frac{y}{x1}\right).p\left(\frac{y}{x2}\right)$ and so on. Equation (6) can be used to combine two different sources of information:

$$P(x1, x2|xy) = P\left(\frac{x1}{y}\right)P\left(\frac{x2}{y}\right) \qquad (6)$$

2.3.3 J48

Decision tree was introduced by [27]. Decision Tree is the common type of classifier used to manage databases for supervised learning. One type of decision tree, J48 creates Univariate Decision Trees that provide predictions about new unlabeled data. J48 uses attribute correlations based on entropy and acquisition of information for each attribute [28]. Various fields of research such as information extraction, data mining, text mining and pattern recognition have used this technique. It is capable of handling various types of input data such as nominal, textual and numerical. The J48 decision tree is an extension of the ID3 algorithm, but it has advantages over ID3. By using a depth-first and divide-and-conquer approach, J48 is able to build small trees.

The decision tree consists of several elements such as the root node, internal node and leaf node. Internal nodes represent the conditions of the parameter values to be tested. A decision tree is built from the top-down root node by partitioning data into subsets that contain the same (homogeneous) instance. The ID3 algorithm calculates sample homogeneity using entropy. If the sample calculated is truly homogeneous, then the entropy is zero and if the sample is evenly divided, then the entropy is one. Entropy shows the amount of information possessed, which means that the higher the entropy, the more information content. A decision tree is constructed by calculating two types of entropy using the rules below:

a) Entropy using the frequency table of one attribute: It can be measured by:

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i \qquad (7)$$

b) Entropy using the frequency table of two attributes:

$$E(T,X) = \sum_{c \in X} P(c)E(c) \qquad (8)$$

Information gain reveals the importance of features or attributes, and determines which attributes are most important for distinguishing between classes that are known. This information is also calculated from training data. The Information gain helps in choosing the best separation; if it has a high value then this split is good, and if the value is low, then the split is not good enough. Information gain is calculated by data obtained from entropy:

Information Gain = entropy (parent) – [average entropy (children)]

2.3.4 Random Tree

Random Tree is a supervised Classifier. Random tree is an ensemble learning algorithm that produces many individual learners. This algorithm uses the idea of bagging to build random data sets in building decision trees. Each code is separated using the best separation among all variables in the standard tree. Each node is split between predator parts randomly selected at that node. This algorithm has been introduced by Leo Breiman and Adele Cutler to deal with classification and regression problems.

The classifier will answer the average response for all trees in the forest in the regression case. Basically, Random Trees is a combination of two algorithms that exist in Machine Learning: a single tree model combined with random forest thinking. Random Forests have been shown to improve the performance of a single decision tree where two randomization methods can create tree diversity [19, 29]. The first way, the training data is sampled by replacing every single tree. Secondly, in calculating the best split for each node, only a random subset of all attributes are considered and the best split of the subset is calculated when growing a tree.

2.3.5 LMT (Logistic Model Trees)

LMT is a decision tree based algorithm that adapts logistic regression to DT induction or a combination of two methods in a single tree structure [30, 31]. LMT consists of a simple DT structure with logistic regression functions in the leaves. The tree structure is almost similar to the C4.5 algorithm. The probability of classification is estimated by the logistic regression function in the LMT structure [30]. The LogitBoost algorithm as an LMT implementation, proposed by Friedman et al. [32] to create logistic regression functions at each tree node. When the logistic regression function is matched to a node using the LogitBoost algorithm, cross-validation is used to determine the number of iterations that are run only once, and use the same number in all trees [33].

Logistic regression from the parent node is passed on to the child node at each split. All the parent models are accumulated as the last model in the leaf node thus creating probability estimates for each class. After growing all the trees, pruning is applied to reduce tree size and increase the generalization of the model [34]. The additive model of the least squares fit *(Fj(x))* uses the LogitBoost Algorithm for the data provided for each class *j*, which has the following form [35]:

$$F_j(x) = \beta_0 + \sum_{i=1}^{T} \beta_i x_i \qquad (9)$$

Where T is the number of features, β_0 is the intercept and βi is the i^{th} component coefficient in the x observation vector. The posterior probability *p(j|x)* for class *j* is modeled by linear logistic regression through a linear function inside x and ensures that all of them are number one and remain in [0,1]. The model follows the following form [35, 36]:

$$F_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_j(x)}}, \sum_{k=1}^{J} F_k(x) = 0 \qquad (10)$$

2.4 Feature Extraction

In increasing the effectiveness of all data mining algorithms and data classification performanse generally uses feature selection procedures [36]. Dataset contains many features which are not all important. Some of the features available are redundant and irrelevant. Redundant features often do not provide additional information, while irrelevant features provide useless information related to the context of the data.

In selecting a subset of the original features used feature selection based on specific criteria and to reduce the dimensions used techniques that are often applied in data mining procedures [37]. The feature selection process is also used to reduce the number of features by eliminating redundant, irrelevant, or noise features. In addition, irrelevant features can increase the complexity of the model and the convergence time for a good model structure. Good feature selection can accelerate the learning process and modeling, improve the accuracy and quality of learning and provide a better understanding of existing models [38].

The evaluator attribute based on metrics is used to rank all features. WEKA has provided this evaluator using the "cfs_subset_eval" correlation subset feature extraction method [39, 40]. In evaluating the importance of a subset of attributes, this method considers the predictive ability of entities of each feature with an existing level of redundancy. Local prediction methods are used to identify, predict and add attributes with the highest correlation of class iterative [41].

Evaluation of missing attribute values is treated as a separate value. The class used by the dataset are nominal and numeric data, and the attributes are non-unary. There are six attributes selected for the anomaly dataset in the order shown in table 2. This attribute is used to handle the misuse attacks shown in table 3.

Table 2: Feature derived for anomaly attacks

Feature	Attribute Name
39	Dst-host-srv-serror-rate
36	Dst-host-same-src-port-rate
33	Dst-host-srv-count
32	Dst-host-count
28	Srv-rerror-rate
12	Logged-in

Table 3: The Features derived for minute attacks

Feature	Attribute Name
2	Protocol-type
3	Service
5	Src-bytes
6	Dst-bytes
17	Num-file-creation
23	Count
24	Srv-count
28	Srv-rerror-rate
32	Dst-host-count
33	Dst-host-srv-count
35	Dst-host-diff-srv-rate
36	Dst-host-same-src-port-rate
39	Dst-host-srv-serror-rate

2.5 Snort Rules

The default snort rule cannot detect further attacks or new attacks so a new signature database is required. The proposed system uses sophisticated signature rules. New snort rules have been configured based on the events and patterns of each attack, Back Attack [42, 43] intends to block web servers where attackers make requests with multiple embed URLs using more slashes.The server has processed every request in response to incoming requests, causing server performance to slow down and become unable to process the original request from the client.

In resolving this situation, snort rules are written to counterattack by: (a) limiting the number of source and destination bytes transferred during transmission when the passing protocol is TCP, (b) limiting flag values to prevent clogging of server resources, (c) limit the number of requests to prevent more URL requests from attackers and (d) limit the spoofing rate to 0.9 or 1.

Another type of attack is when an attacker sends a fake SYN with the same source and destination address. The number of srv_count values is set to limit the SYN packet so that spoofing can be prevented. DOS attacks (Neptune or SYN Flood [44, 45] cause an abundance of data structures to store information about each new connection request received. An open TCP connection causes the server to always receive and add notes to the data structure. Its limited size causes the overflow of data structures on the server, so that the data structure cannot receive requests again until the old data structure is released. The proposed snort rules are shown in table 4.

Table 4: Snort Rules for KDD Cup

Method of Attack	Snort Rule
Smurf	Alert to Protocol=icmp, service=ecr_i, flag=SF, src_bytes=1032, dst_host_count=255, dst_host_diff_srv_rate=0
Teardrop	Alert to Protocol=udp/tcp, service=telnet, flag=SF, src_bytes=237, dst_bytes=1540, dst_host=count=255, dst_host_same_srv_rate=0.18
Neptune	Alert to Protocol=tcp, service= private/smtp/telnet, flag=SO, serror_rate=1/0.94, srv_serror_rate=1/0.93
Pod	Alert to Protocol=icmp, service= ecr_i/tim_i, flag=SF, src_bytes=1480, wrong_fragment=1, dst_host_count=255, dst_host_diff_srv_rate=1
Nmap	Alert to Protocol=tcp/udp/icmp, service= private/nmp/telnet, flag=SF/SH, src_bytes=207, same_srv_rate=1/0.5, srv_diff_host_rate=1
Satan	Alert to Protocol=tcp, service=private/telnet, flag=SF/REJ, src_bytes=54, dst_host_count=255, dst_host_same_src_port_rate=0
Portsweep	Alert to Protocol=tcp, service= private/ftp/telnet, flag=SO/RSTR, dst_host_count=255, dst_host_srv_count=2
Ipsweep	Alert to Protocol=icmp, service=eco_i/private, flag=SF/RSTO/REJ, src_byte=8, count=1, dst_host_count=71
Back	Alert to Protocol=tcp, service=http, flag=SF/SH/RSTR, src_bytes=54540/54060, dst_byte=7300/8314, host=2, srv_count=13, same_srv=1/0.8
Spy	Alert to Protocol=tcp, service=telnet, flag=SF, dst_host_diff_srv_rate=0.02
warezmaster	Alert to protocol=tcp, service=ftp/ftp_data, flag=SF, dst_host_count=218, dst_host_srv_count>10
warezclient	Alert to Protocol=tcp, service=ftp/ftp_data, flag=SF/RSTO, src_bytes=>980, dst_bytes>1208, hot>10, dst_host_count=255, dst_host_srv_serror_rate>0.02

Multihop	Alert to Protocol=tcp, service= ftp_data/telnet, flag=SF, src_bytes=1412, dst_bytes=988002, dst_host_srv_count=3, dst_host_same_src_port_rate=1
Imap	Alert to Protocol=tcp, service=imap4, flag=SF/SH, count=4, dst_host_same_srv_rate=1, dst_host_srv_count=1
Ftp_write	Alert to Protocol=tcp, service=ftp/ftp_data/login, flag=SF, src_bytes=676, dst_byte=39445, loggin_in=1
Guess_passwd	Alert to Protocol=tcp, service=telnet, flag=SF/RSTO, src_bytes=125 or 126, dst_byte=179, hot=1, num_failed_login=1
Land	Alert to Protocol=tcp, service=finger/telnet, flag=SO, land=1, src_count=2, dst_host_srv_serror_rate=0.58/0.12
Rootkit	Alert to Protocol=tcp/udp, service=ftp_data/telnet, flag=SF, logged_in=1, dst_host_count=255, dst_host_srv_rate=0
Perl	Alert to Protocol=tcp, service= telnet, flag=SF, logged_in=1, dst_host_srv_count=2, dst_host_diff_srv_rate=0.07
Load Module	Alert to Protocol=tcp, service=ftp/ftp_data/telnet, flag=SF, dst_host_count=6, dst_host_same_src_port_rate=1/0.25
Buffer Overflow	Alert to Protocol=tcp, service=ftp/ftp_data, telnet, flag=SF/RSTO, src_bytes=6247, dst_byte=70529, loggin_in=1, dst_host_same_srv_rate=1
Phf	Alert to Protocol=tcp, service= telnet, flag=SF, src_bytes=51, dst_host_count=255, dst_host_same_srv_rate=1

3. RESULT AND DISCUSSION

3.1 Implementation of the Selected Classifiers

The KDD dataset has 41 features about each data packet that implements various types of classifiers on cable networks. Experiments that have been carried out present a fair testing environment because the training dataset is extracted from 148,753 instances of the four attack groups (DOS, R2L, U2R, and PROBE), normal dataset packets are drawn from 19% of experimental data, and original DOS KDD dataset packets from the highest proportion for DOS attacks taken from 79% of experimental data. In order to create a fair comparison of controls between different classifiers, the test sample was extracted from the original KDD dataset of 77,000 randomly independent samples and not included in the training dataset.

This experiment uses Weka version 3.8 and Intel Core (R) CPU I5-2410M @ 2.30GHz X4 with 4.0GB of available RAM under the Windows 10 platform. The classifications used in this experiment are Bayes (Bayes Network, Naive Bayes) and Decision Tree (Random Tree, LMT, J48). All classifiers and their results are saved to begin a comprehensive study of which classifier has the highest level of accuracy to detect attacks.

3.2 Snort Results

The Snort Rule has been tested in real-time traffic and data sets with attack experiments conducted and detected every day for one week. Data files have been downloaded from experiments on the internal network. The number of packages and attacks detected in one week is shown in table 5. Packets are captured and analyzed using the "Wireshark" tool and the graphical representation of the packets is shown in figure 3.

There are 320 attacks have been detected by snort on Monday. There are a total of 1033 network packets, of which 993 are TCP packets and 40 are UDP packets. In detecting TCP packets, the total time required is 0.0040 milliseconds, while to detect UDP packets, the time required is 0.0005 milliseconds.

Table 5: Attacks detected on daily basis for one week

Day	Protocol	Counts	Number of Packet	Rate (ms)	Attacks Detected
Monday	UDP	993	1033	0.0040	320
	TCP	40		0.0005	
Tuesday	UDP	1120	1800	0.0060	250
	TCP	680		0.0030	
Wednesda	UDP	1340	1407	0.0140	275
	TCP	67		0.0009	
Tuesday	UDP	1775	2065	0.0045	210
	TCP	290		0.0010	
Friday	UDP	1919	2059	0.0045	247
	TCP	140		0.0009	
Saturday	UDP	1750	1800	0.0088	276
	TCP	50		0.0004	

Sun day	UDP	2200	3300	0.0089	380
	TCP	1100		0.0053	

Snort rules are created and tested in real time traffic, with various security breach attempts such as TCP packet flooding [46, 47] to test system efficiency. The website that is tested will produce more TCP packets during communication, and the rules that are tested on this packet are for capturing and detecting network threads.

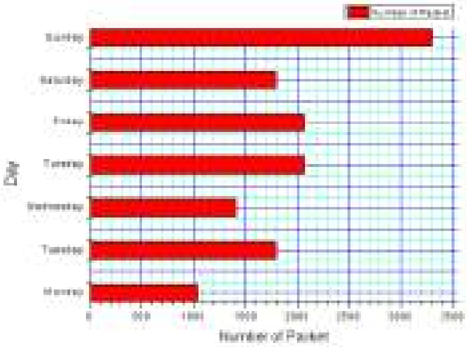

Figure 3: Packets Number taken using Snort

The total number of attacks detected using the snort rule used is shown in figure 4. Snort rules and performance have been observed and tested in one week period.

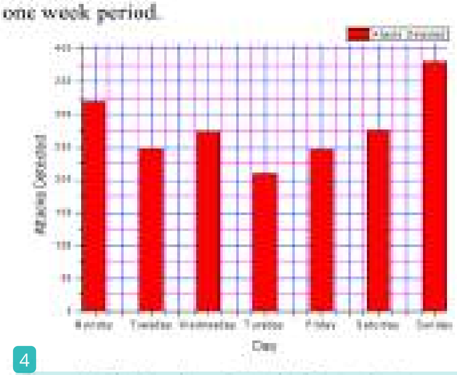

Figure 4: The Number of Attacks Detected using snort rules

3.3 Classification Algorithm Results

WEKA is used in many common data collection techniques such as data cleaning, classification, grouping, data pre-processing, regression, visualization, and feature selection. In this case, WEKA as a data mining technique for conducting many classification experiments with NSL-KDD as a dataset.

WEKA tools are also used to evaluate algorithms and compare results with those obtained using the same values and parameters. Parameters such as the accuracy of the technique in detecting attacks that affect NSL KDD dataset training and testing are used as a comparison. In addition, various different classifiers were also tested such as Bayes (Bayes Network, Naive Bayes) and Decision Tree (Random Tree, LMT, J48) using NSL KDD dataset.

This section describes all the parameter values used by the classifier in the experiment. All selected classifiers were tested with 77,000 random instances randomly from the KDD dataset.

The parameters used are tested by the Bayes Network Classification: search technique=K2 algorithm and search techniques estimator value = simple estimator, while the parameters used to test the Random Tree Classifier: seed = 1 and Min variance = 0.002. Parameters used to test Decision table classifier: cross value = 1 and search techniques best first, while the parameters used to test J48 classification: numFolds = 2, confidence factor = 0, seed = 1, ascended sub tree = true, unsuned = False, and fallen tree = true.

In figure 5, lists statistical values that achieved in experiments and it can be seen that Random Tree classifier achieves the highest Kappa statistic with rate equals to 0.885 and the lowest Kappa statistic with Bayes Network classifier with rate equals to 0.836.

In figure 6 and 7, records weighted average for true positive (TP), false positive (FP), precision and ROC Area for each classifier selected for experiment. Figure 8 presents the experiment records of accuracy rate, correctly classified, and incorrectly classified. The highest accuracy is achieved by J48 Classification.

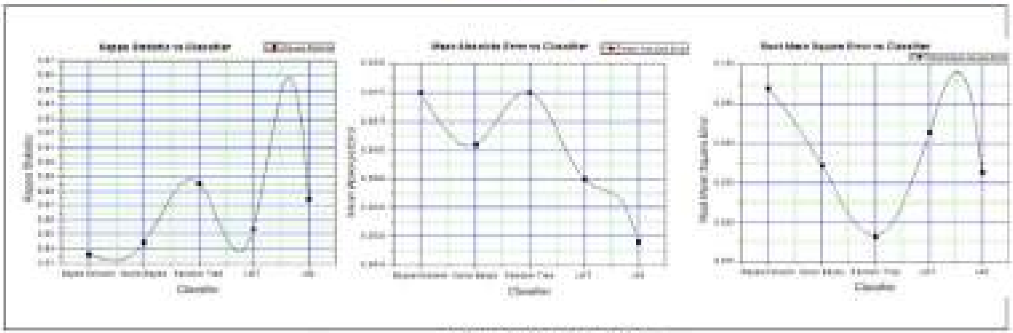

Figure 5: Value Statistics

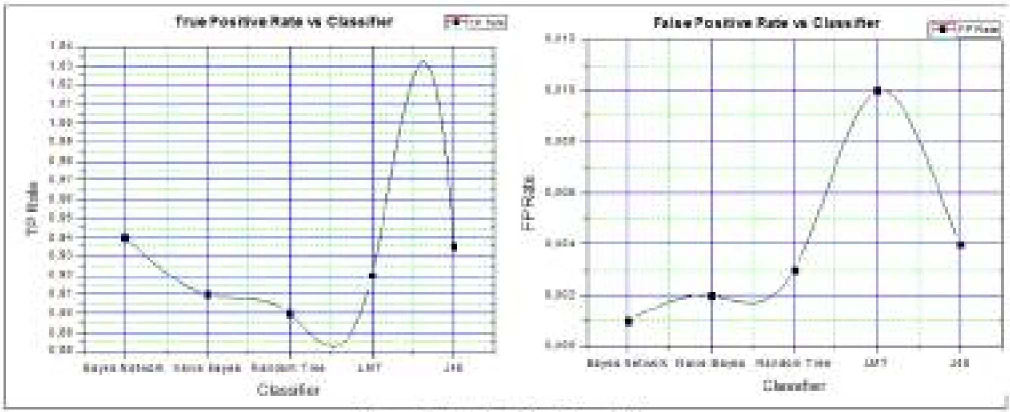

Figure 6: Weighted for TP and FP

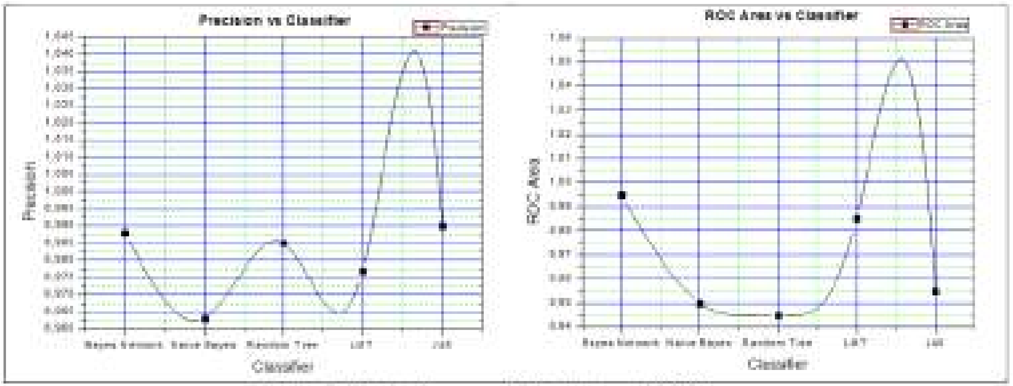

Figure 7: Weighted average for Precision and ROC Area

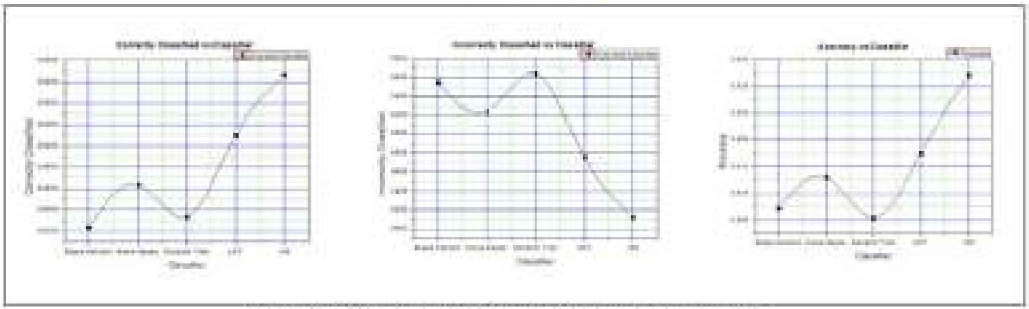

Figure 8: Correctly, Incorrectly Classified and Accuracy rate

10-fold cross validation and the test set provided is also used to determine the accuracy of the prediction model. In this method, 10 different samples are randomly divided, of which 4 samples are used for testing purposes, while the rest are used for training purposes.In addition, the authors provide a classifier test set to investigate its performance in terms of predicting the class of the sample set taken from the file. Various parameters are used to determine the accuracy and robustness of the approach. Proving the best accuracy and performance compared to other techniques is shown by the J48 algorithm.

3.3.1 Testing using 10-fold cross validation

This research has used WEKA to investigate the algorithm and the results are compared using standard algorithms with the same experimental conditions.

The authors have compared the results using many algorithms such as Bayes (Bayes Network, Naive Bayes) and Decision Tree (Random Tree, LMT, J48) with the same experimental settings.

Table 6 compares the accuracy of intrusion detection for all the algorithms researched. This experiment has used a complete NSL KDD (KDDTrain +) training dataset that has 41 features and 10-fold cross validation test. it appears that BayesNet showed the lowest accuracy of 75.38% for all 41 features and the J48 algorithm shows the highest detection accuracy of 99.81%.

Table 6: Analysis of accuracy using 10-fold cross validation.

Detection Accuracy	
Bayes Network	75.38
Naive Bayes	91.77
Random Tree	98.34
LMT	76.44
J48	99.81

3.3.2 Testing without feature selection technique

The authors have compared several algorithms such as Bayes (Bayes Network, Naive Bayes) and Decision Trees (Random Tree, LMT, J48) with the same parameters and values assisted by the supply test set.

The comparison results of detection accuracy for all algorithms used in IDS are shown in table 7. The dataset used is NSL KDD (KDDTrain +.) and Full Test NSL KDD (KDDTest +.) which has 41 features and supplied test sets.

Table 7:. Analysis of accuracy using supplied tests and Full test NSL KDD dataset

Detection Accuracy	
Bayes Network	74.38
Naive Bayes	77.71
Random Tree	85.31
LMT	79.43
J48	91.81

Table 7 shows that the highest detection accuracy of 91.81% for all 41 features produced by the J48 algorithm; while the lowest accuracy is shown by Bayes Network (74.38%).

Table 8: Analysis of the accuracy using supplied tests and Test-21 NSL KDD dataset

Detection Accuracy	
Bayes Network	52.34
Naive Bayes	55.65
Random Tree	68.98
LMT	31.34
J48	65.84

In the table 8, the authors compare the accuracy of intrusion detection from the algorithm with many other data mining techniques. Here, the authors have compared algorithms such as Bayes (Bayes Network, Naive Bayes) and Decision Tree (Random Tree, LMT, J48) under the same values and parameters.

4. CONCLUSIONS

Real time internet traffic has been captured firstly using Snort software which is a packet capturing tool and rule based classification. The pattern is matched with the existing signature base of the snort rule. Existing snort rules and tested systems prove that the proposed rules are able to detect more accurately. After that, Internet traffic is classified using five machine learning classifiers. This work phase begins with the preparation phase of experiment and completes the independent random test data from KDD.

Among the several classification techniques used (Bayes Network, Naive Bayes, Random Tree, LMT, J48), all types of attacks from the KDD dataset (DOS, R2L, U2R, and probing) can be detected and classified by the J48 algorithm with the highest accuracy. Results show that J48 gives accuracy of 99.81% in experiment has used a complete NSL KDD (KDDTrain +) training dataset that has 41 features and 10-fold cross validation test. Also in the dataset used is NSL KDD (KDDTrain +.) and Full Test NSL KDD (KDDTest+.) which has 41 features and supplied test sets, the highest detection accuracy of 91.81% for all the complete records of the KDD dataset (41 features) produced by the J48 algorithm.

For further research related to this data set in the future, a data engineering phase is needed focusing on attributes that are able to achieve the highest accuracy for used classifiers.

REFERENCES

[1] Y. Zhu, "A Data Driven Educational Decision Support System," *International Journal of Emerging Technologies in Learning*, Vol 13, No 11, 2018.

[2] N. Huy and C. Deokjai, "Application of Data Mining to Network Intrusion Detection: Classifier Selection Model, *"Challenges for Next Generation Network Operations and Service Management*, pp. 399--408, 2008.

[3] S. Paliwal and R. Gupta, "Denial-of Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm," *International Journal of Computer Applications*, vol. 60, no. 19, pp. 57--62, 2012.

[4] Witten, I.H., Frank, et. al., "Data Mining: Practical Machine Learning Tools and Techniques",. *Morgan Kaufmann*. 2016.

[5] Weiming H., Wei H., et. al., "AdaBoost-based Algorithm for Network Intrusion Detection", *IEEE Trans. Syst. Man Cybern. B Cybern.* 38, 577–583, 2008.

[6] Li W., et. al.,"A New Intrusion Detection System based on KNN Classification Algorithm in Wireless Sensor Network", *J. Electr. Comput. Eng. 1–7* (2014). doi:10.1155/2014/240217, 2014.

[7] Ahmed M., et. al., "A Survey of Network Anomaly Detection Techniques", *J. Network. Computer. Appl.* 60, 19–31, 2016.

[8] Vasudevan, A.R., E. Harshini, and S. Selvakumar, "SSENet-2011: a Network Intrusion Detection System Dataset and its Comparison with KDD CUP 99 Dataset", *IEEE, 978-1-4577-1088-9*, 2011.

[9] Claude Turner et al., "A Rule Status Monitoring Algorithm for Rule-Based Intrusion Detection and Prevention Systems, Complex Adaptive Systems, Conference Organized by Missouri University of Science and Technology - Los Angeles, CA, *Procedia Computer Science 95*, 361 – 368, 1877-0509, doi: 10.1016/j.procs.2016.09.346, 2016.

[10] Anna L. Buczak, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection", 10.1109/COMST.2015.2494502, *IEEE Communications Surveys & Tutorials*, 1553-877X, 2015.

[11] Syed Rizvi et al,"Advocating for Hybrid Intrusion Detection Prevention System and Framework Improvement", Complex Adaptive Systems, Conference Organized by Missouri University of Science and Technology 2016 - Los Angeles, CA, *Procedia Computer Science 95*, 369 – 374, doi: 10.1016/j.procs.2016.09.347, 2016.

[12] Mohsen Rouachedab, Hassen Sallay, "An Efficient Formal Framework for Intrusion Detection Systems", *The 2nd International Symposium on Frontiers in Ambient and Mobile Systems(FAMS)*, Procedia Computer Science, 968–975, 1877-0509, doi: 10.1016 / j.procs. 2012. 06.132, 2012.

[13] SUN et al., "Anomaly Detection Based Secure In- Network Aggregation for Wireless Sensor Networks", *IEEE System Journal*, Vol. 7, No. 1, March 2013, 13-25, 1932-8184, Digital Object Identifier 10.1109/JSYST.2012.2223531, 2013.

[14] A. Saidi et al., "The functional of A Mobile Agent System to Enhance DoS and DDoS Detection in Cloud", *International Journal of Applied Engineering Research*, ISSN:0973-4562, Vol. 11, No. 6 pp 4615-4617, 2016.

[15] Blum, Avrim L. & Pat Langley, "Selection of Relevant Features and Examples in Machine

Learning", *Artificial Intelligence*, 97(1-2), 245–271, 1997.

[16] D. Gaikwad, S. Jagtap, K. Thakare and V. Budhawant, "Anomaly based Intrusion Detection System using Artificial Neural Network and Fuzzy Clustering", *International Journal of Engineering*, vol. 1, no. 9, 2012.

[17] H. Nguyen and D. Choi, "Application of Data Mining to Network Intrusion Detection:Classifier Selection Model," *Challenges for Next Generation Network Operations and Service Management*, pp. 399--408, 2008.

[18] M. K. Lahre, M. T. Dhar, et. al., "Analyze different approaches for IDS using KDD 99 data set," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 1, no. 8, pp. 645--651, 2013.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5--32, https://doi.org/10.1023/A:1010933404324, 2001.

[20] R. Kohavi and G.H. John, "Wrappers for Feature Subset Selection". *Artificial Intelligence.97* (1-2), 273- 324. 1997.

[21] Natttawat Khamphakdee, et. al.,"Improving Intrusion Detection System based on Snort Rules for Network Probe Attack Detection", *International conference on information and communication technology*, IEEE, 2014.

[22] Blum, Avrim L. & Pat Langley, "Selection of Relevant Features and Examples in Machine Learning, "*Artificial Intelligence*, 97(1-2), 245–271, 1997.

[23] M. Ali. Aydin, A. Halim Zaim and K. Gokhan Celyan, "A Hybrid Intrusion Detection System Design for Computer Network Security", *Computer and Electrical Engineering 35*, 517-526, 2009.

[24] L. Fausett and L. Fausett, "Fundamentals of Neural Networks: Architectures, Algorithms,and Applications", *Prentice-Hall*, 1994.

[25] J. Jorgenson, et. al., "A hierarchical Anomaly Network Intrusion Detection System Using Neural Network Classification," in *Workshop on Information*, 2001.

[26] K. Gurney, "Introduction to Neural Networks", *CRC press*, 1997.

[27] Breiman, Leo, et. al., "Classification and Regression Trees", *Wadsworth, Belmont, CA*, 1984.

[28] N. Bhargava, et. al., "Decision Tree Analysis on J48 Algorithm for Data Mining", *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, 2013.

[29] Amit. Yali, et. al., "Shape Quantization and Recognition with Randomized Trees" *(PDF). Neural Computation 9* (7): 1545–1588. 1997.

[30] Landwehr N, Hall M, Frank E. "Logistic Model Trees". *Machine Learning*. 59:161–205. 2005.

[31] Sumner M, et. al., "Speeding up logistic model tree induction. In: Jorge A, Torgo L, Brazdil PB, Camacho R, Gama J, editors". *Knowledge Discovery in Databases: PKDD* 2005. Heidelberg: Springer; p. 675–683. 2005.

[32] Friedman J, Hastie T, Tibshirani R. "Additive Logistic Regression: A Statistical View of Boosting (with discussion and a rejoinder by the authors)." *Ann Stat*. 28:337–407, 2000.

[33] Witten IH, et. al, "Data mining: practical machine learning tools and techniques", *3rd ed. Burlington (MA):* Morgan Kaufman. 2011.

[34] Doetsch P, Buck C, et al., "Logistic Model Trees with AUC Split Criterion for the KDD Cup 2009 Small Challenge". *Machine Learning Res Proc*. 7:77–78. 2009.

[35] Shoombuatong W, et al., "HIV-1 CRF01_AE Coreceptor Usage Prediction using Kernel Methods based Logistic Model Trees", *Computer Biology Medicine*. 42:885–889, 2012.

[36] Ravage U., Marathe N., et al. "Feature Selection based Hybrid Anomaly Intrusion Detection System Using Kmeans and RBF Kernel Function". *Procedia Comput. Sci. 45*, 428–435, 2015.

[37] De la Hoz E., Ortiz, A., et al. "PCA Filtering and Probabilistic SOM for Network Intrusion Detection. *Neurocomputing 164*, 71–81, 2015.

[38] Najafabadi M.M., et. al.,"Evaluating Feature Selection Methods for Network Intrusion Detection with Kyoto Data", *Int. J. Reliab. Qual. Saf. Eng*. 23(01), 1650001, 2016.

[39] R.E.Mohammad, Menar S., et al. "Intrusion Detection Using Data Mining Techniques", *IEEE*, 2010.

[40] G. V. Nadiammai and M. Hemalatha, "Handling Intrusion Detection System using Snort based Statistical Algorithm and Semi-supervised Approach", *Research Journal of Applied Sciences, Engineering and Technology 6* (16): 2914-2922, 2013.

[41] Matthew V. Mahoney, "Network Traffic Anomaly Detection based on Packet Bytes", *ACM*, 2003.

[42] Matthew V. Mahoney and Philip K. Chan, "Learning Rules for Anomaly Detection of Hostile Network Traffic", *Florida Institute of Technology*, Melbourne, FL 32901, 2003.

[43] Vinod Kumar and Om Prakash Sangwan, "Signature-based Intrusion Detection System using SNORT", *International Journal of computer application & information technology*, 2012.

[44] M. Naga Surya Lakshmi, Y Radhika. "Effective Approach For Intrusion Detection Using Ksvm And R", *Journal of Theoretical and Applied Information Technology*. Vol95. No.17, ISSN: 1992-8645, 15th September 2017.

[45] Summary Thaseen and Aswani Kumar, "Intrusion detection model using fusion of PCA and optimized SVM", *IEEE*, 2014.

[46] Naila Belhadj Aisa, Mohamed Guerroumia, "Semi-Supervised Statistical Approach for Network Anomaly Detection", *The 6th International Symposium on Frontiers in Ambient and Mobile Systems*, Procedia Computer Science 83 1090 – 1095, 1877-0509, 2016.

[47] Divya and Surendra Lakra, "SNORT: A Hybrid Intrusion Detection System using Artificial Intelligence with a Snort", *International Journal Computer Technology & Application*, Vol 4(3), 466-470, 2013.

NETWORK INTRUSIONS CLASSIFICATION USING DATA MINING APPROACHES

ORIGINALITY REPORT

18%	19%	19%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	zuscholars.zu.ac.ae Internet Source	6%
2	Submitted to Arab Open University Student Paper	4%
3	link.springer.com Internet Source	4%
4	www.ripublication.com Internet Source	4%

Exclude quotes	On	Exclude matches	< 3%
Exclude bibliography	On		