Adaptive Hybrid Model for Network Intrusion Detection and Comparison among Machine Learning Algorithms

Md. Enamul Haque and Talal M. Alkharobi

***Abstract*—In this paper, we propose a novel method using ensemble learning scheme for classifying network intrusion detection from the most renowned KDD cup dataset. We have shown that reducing the dimensionality of the large dataset provides most accurate detection. Additionally, several machine learning algorithms are used to generate the accuracy metrics and analyzed further for proper comparison. Our approach found out that this algorithm outperforms all other learning techniques. Our goal is to analyze the network intrusion data and find out the best components and use them for the attack analysis. This scheme can be used in parallel with the intrusion detection system to augment its prediction performance for the future data packets. Empirical results show that the input dimensionality reduction can provide lightweight intrusion detection system that can be embedded with the vulnerable system for generating correct classification with significance improvement in execution time.**

***Index Terms*—Network intrusion detection, Random Forest, PART, Naive Bayes, machine learning.**

I. Introduction

The recent advancement of network technologies has opened tremendous scope for new and improved types of vulnerabilities for the systems used all over the world. Enterprises and individuals struggle with their valuable, confidential and personal information due to this unbounded connectivity among hundreds of thousands computers. Although internet has brought significant improvements in our daily lives, it has added several cautions as well. If we want to look deep into the infrastructure of contemporary business enterprises or companies, the information technology would appear as the most prominent components which include computers, networking devices, multimedia devices, complex servers etc. In most cases organizations are highly dependent on these technology driven processes, for example to maintain employee records, to keep financial transaction records, to keep client information, and to store organization specific confidential data etc. Our day starts with checking emails, Facebook posts, tweets from twitter, checking online accounts, shopping online through credit cards, chatting with friends etc. All of these communication requires individual information exploitation over the internet and very vulnerable to be captured by other malicious users.

This valuable information can be captured from the network with different malicious software and processes developed by numerous attackers. Their prime goal is to capture the in between communication data and dig out relevant information from there for further usage for benefits. By the time this paper is being written, IT security vendor Kaspersky Lab has released a real-time interactive map of online malware threats. It gives us a glimpse of how the whole world is being impacted day by day with network intrusion.

Thus we need strong surveillance systems in the form of real time intrusion detection systems [1] to protect our valuable information from these attacks. Most of the attacks are originated from unauthorized access to the computer systems or computer networks. This is defined as intrusion where outsiders try to gain access into a system without legal permission. So, we need intrusion prevention systems which are defined as the processes those encourage the system to react against both present and foreseeable attacks. There is a strong association between these two processes in terms of responses. Firstly, intrusion detection process acts as passive where attacks are detected. Secondly, intrusion prevention systems do not let the system to be attacked. It analyzes the network traffic iteratively to filter out any suspicious packets to ensure prevention from malicious attacks. Our intrusion detection process is defined in such a way so that any upcoming attacks can be prevented from the previously detected information.

We have used Naive Bayes, Random Forest and PART along with feature reduction schemes to classify the NSL KDD dataset [2] into normal and attacked types. Both training and testing data is used along with 10-fold cross validation. Additionally, the features were reduced to some level to have better accuracy using principal component analysis and ranking algorithms. The NSL KDD dataset has four basic types of attacks defined in the dataset [3]. Our approach can satisfy the accuracy metric to detect any abnormalities within the captured network data by the analyzers. Finally, we compare the performance among various data mining algorithms in terms of prediction accuracy by tuning the specific parameters for those algorithms.

The remainder of this paper is organized as follows. In Section II, we mentioned a concise overview of the related work done in this area. We explained intrusion detection method briefly in Section III. In Section IV, we describe our experimental setup mentioning how hybrid machine learning algorithms and feature selection comes into action. Section V provides detail information about the dataset used for the experiment. In Section VI, we demonstrate our experimental

Manuscript received July 14, 2014; revised October 24, 2014. This work was supported in part by the Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Kingdom of Saudi Arabia.

Md. Enamul Haque is with King Fahd University of Petroleum & Minerals, Dhahran, 31261, Kingdom of Saudi Arabia (tel.: +966-50-2389368; e-mail: g201204920@kfupm.edu.sa, enamul_cse@yahoo.com).

Talal M. Alkharobi is with the Department of Computer Engineering, King Fahd University of Petroleum & Minerals, Dhahran, 31261, Kingdom of Saudi Arabia.

DOI: 10.7763/IJMLC.2015.V5.476

results and evaluation details. Finally in Section VII, we conclude and suggest future direction of our work.

II. Related Work

We will discuss contemporary work relevant to our research in this section. Kim *et al.* [4] proposed a new hybrid intrusion detection model that integrates both misuse and anomaly detection model hierarchically. They decomposed the normal data into smaller subsets using misuse detection model. Later, one-class SVM models were built from those decomposed data to have precise behavior from the normal data profile. They evaluated the model using the most used NSL-KDD intrusion detection dataset. They claim that the model outperforms conventional methods in terms of detection rate and low false positive rate. Additionally, the training and testing time was depreciated by 50% and 60% respectively when compared to the traditional methods.

A new feature generation approach called four-angle-star is proposed by Luo *et al.* [5]. Their aim was to evaluate the distance between samples in that star approach where 5-classes were available. They generated several numeric features from KDDcup99 data with the help of four angle star image. The FASVFG classifier achieves high generalization accuracy.

Fung *et al.* [6] proposed a Bayesian trust management model for distributed intrusion detection networks. Their model ensures both trust estimation and its confidence. They mentioned two important factors those are added in their research. First, the model can identify between insider attacks and outsiders attacks and can be scalable for large networks. Dirichlet density function was used in the trust modeling field to ensure high level of confidence while estimating trustworthiness of IDS.

Alsubhi *et al.* [7] mentioned intrusion detection and/or prevention systems (IDPSs) as crucial defensive mechanism to get rid of the attacks by the intruders or hackers. Their study reveals that the IDPS configuration may impact the network performance adversely in terms of end-to-end delay and packet loss. They proposes one analytical queuing model using embedded Markov chain that can analyze the performance of IDPS and evaluate its performance on the network. Their approach outperforms other contemporary works and can achieve the security tradeoffs along with network quality of service.

Haines *et al.* [8] discussed about three variant of the performance evaluations performed on 1998 and 1999 DARPA dataset which was offline. The first variant was called LARIAT (Lincoln Adaptable Real-time Information Assurance Test bed) which can help the researchers configure and run real time intrusion detection system. It also helps to make correlation test with strong background traffic. Secondly, they provided with scenario datasets that overcomes the limitations of the past evaluations. Thirdly, they have analyzed the DARPA 1999 dataset extensively that can help model new type of intrusion detection systems.

Neill *et al.* [9] proposed two different scalable intrusion detection systems architecture. One model uses large scale parallel signature pattern to introduce anomalous intrusion alerts. They claim that this system acts robustly for the distributed hosts systems. The intrusion data is collected from a central host, which is then clustered and mapped to one or more controlled hosts. These control hosts manages the parallel sequence processing for the clusters on the broadband embedded cluster Set-Top Boxes. Similarly, they defined another intrusion detection process for the multiple network attack IDS.

Kumar *et al.* [10] introduced a unique agent based process that collects network data and apply Artificial Immune System to detect any intrusion. This method also helps the data to travel over more secured channel instead of using the intrusion prone channels. Their systems run normally without any misbehavior and provide promising result.

Intrusions detection is not only the concern for the current scenario. The concerned team has to take action based upon the detection alarm. The action can be triggered manually or via automated signaling from the initial intrusion alert. Saman *et al.* [11] has proposed a new technique based on the intrusion detection and response scenario, which they named response and recovery engine (RRE). Their model uses a game theoretic response strategy based on two-player Stackelberg stochastic games. The RRE applies attack response trees (ART) and concludes with some sort of attack prediction. They also employed fuzzy logic theory to calculate network-level security measures.

Weiming *et al.* [12] mentioned that the current intrusion detection systems lack to keep in pace with the frequently changing of the network environment. The authors mentioned two variant of Adaboost process. In first Adaboost version, they used decision stumps as weak classifiers. Secondly, an improved and on line version of Adaboost is proposed with the Gaussian Mixture Model (GMM) as weak classifier. They also used Particle Swarm Optimization (PSO) and Support Vector Machine (SVM) for global detection model.

Monowar *et al.* [13] mentioned several familiar aspect of network anomaly detection. They have compared and discussed about large number of network anomaly detection tools.

Darren *et al.* [14] introduced the general testing approach called Mucus which analyzes the data captured from the system. The authors used cross testing experiments with both an open-source and commercial tool. An evasion attack was also mentioned due to the test result.

Shiri *et al.* [15] proposed parallel version of an intrusion detection system to improve the performance of signature based intrusion detection system. They also mentioned that two signature based intrusion detection systems run parallel with some portion of packets and a subset of rules. The processing time of the traffic improves.

Alghamdi *et al.* [16] proposed to use PCA (Principal Component Analysis) to reduce the high dimensionality of the dataset. They used batch back propagation neural network for the classification among the dataset. They measured the RMSE (Root Mean Squared Error) at the training phase and set some acceptance range of 0.1. They used JUNE for implementing modular neural network, JoonePAD for graphical PCA and RMSE during learning phase.

III. Intrusion Detection System

Intrusion detection system is almost 20 years [17] old topic for the research community. In general, intrusion detection refers to any unauthorized access into system which can be tracked by comparing with the authorized access behavior pattern data. Additionally, if any access violates either confidentiality, integrity or accessibility of any system then that access is not normal. In most cases, intrusion detection systems are not treated as the primary shield of defense mechanism. Usually, it augments the efficiency of the other primary level security systems like access control, firewall, authentication etc. Aurobindo Sundaram [18] divided intrusions into six types: Attempted break-ins, Masquerade attacks, Penetrations, Leakage, Denial of service, and malicious use. We can classify this intrusion detection system (IDS) into two major categories based on deployment: host based or network based.

Host based intrusion detection systems (HIDS) mostly serve as an internal agent to the system. They monitor and analyze the inter system behaviors, for example, some word processor has unexpectedly started to modify the system password files and it is detected by the detection system. These detectors have access to the internal memory, log files, file system etc. Above all, host based intrusion detection systems try to keep the operating system integrity either from internal or external modification.

Network based intrusion detection systems (NIDS) stays in active mode all the time for the system to be monitored [19]. They monitor and analyze the network data coming into the system. They try to find the anomalous behavior from the incoming packets, for example, if some attacker tries to scan ports of a system, the NIDS can detect a huge number of TCP connection requests for different port scan over a very short period of time. Then it can easily detect that someone is using port scans over the network. The NIDS have to be intelligent enough to handle large amount of data and differentiate among different patterns. The data can be either continuous or discrete. We can further classify NIDS into two categories: misuse-based NIDS and anomaly-based NIDS. Misuse-based NIDS usually searches for known intrusion patterns and anomaly-based NIDS searches for both known and unknown patterns. Recent researches are mostly concerned about anomaly-based NIDS, thus our research focuses on this class.

Anomaly based systems learn from the network and generates rules for classifying the activities as either normal or anomalous. This is opposite to signature based systems which can detect attacks for which some signature value has assigned previously. Before classifying the network packets as normal or anomalous, the system is required to learn about the normal behavior. This can be performed in different ways, mostly using artificial intelligence and machine learning techniques. Another version of anomaly based detection is strict anomaly based detection method in which strict mathematical models are used for tracking any deviation from normal traffic class. Past researches reveal that anomaly based intrusion detection has several flaws, namely a high false positive rate and misjudging a correct data packet as attack packet. PAYL [20] and MCPAD [21] have tried to address these issues. In this research we will also focus on reducing these issues using both supervised and unsupervised anomaly-based intrusion detection.

A. Supervised Anomaly Detection

When both normal and anomaly data is labeled for training and to make a predictive model, we call it supervised anomaly detection [22] model. In this case, any unseen data instances are compared against the model to decide which class it belongs to. But, there are some issues found related to the imbalance of normal and anomaly dataset. This was solved in [23] using machine learning and data mining techniques. Another issue is to find the optimized labels for the anomaly class which is really challenging. It was addressed in [24] using artificial anomaly injection in the normal dataset to produce labeled training data set.

B. Semi Supervised Anomaly Detection

On the other hand, semi supervised anomaly detection techniques use only one labeled class for normal data for the training purpose. In most cases normal class is modeled due to the complexity of modeling anomaly class. For example, in road accident detection, an anomaly scenario would add the occurrence of an accident, which is complex to model. There are several systems as well those rely on using the training data set as anomaly set only. These systems lack predictability because it is not easy to create some anomalous dataset that will cover every possible abnormal behavior within the system.

C. Unsupervised Anomaly Detection

Unsupervised anomaly detection techniques do not require any training data. These techniques assume that the frequency of normal data is far greater than the frequency of anomalous data. Thus, the techniques can end up providing high false positive rate if the primary assumption violates.

IV. Proposed Model

We already know about the classification of intrusion detection system, and we are concerned about the network anomaly detection techniques. Our proposed model lies into both supervised and semi supervised anomaly detection technique as it uses the training dataset consisting of both normal and anomalous class. Initially, our proposed model uses different network analyzer to collect data from the network and stores in MySQL database for further pruning. The database is filtered with only unique dataset to reduce the probability of setting much emphasize on particular subset of data. This data collection and pruning is being done simultaneously to make the system more adaptive. Once a cluster of data is filtered, it is sent for preprocessing in another module. Preprocessing step makes the balance between normal and anomaly class data and selects the feature set to be used. Then the data is processed in feed forward neural network for classifying. After the initial iteration, the data set is checked based on the weight of the classification. The features those were contributing in the earlier iteration for deciding the class label are identified using principal component analysis (PCA). In the next iteration, those features get more importance while classification. Thus the

system learns the environment better and produces more accurate outcome as it gets older.

Algorithm 1. Hybrid algorithm for adaptive network intrusion detection.

1. Capture network data.
2. D ← Stored data from database.
3. N ← all feature set.
4. th← threshold value.
5. **for** $i = 1 \rightarrow n$ **do**
6. F = F - F_i
7. ac = calculateAccuracy(F)
8. **if** ac ≤ th **then**
9. break
10. **end if**
11. **end for**
12. Apply learning algorithms.
13. Classified Data.

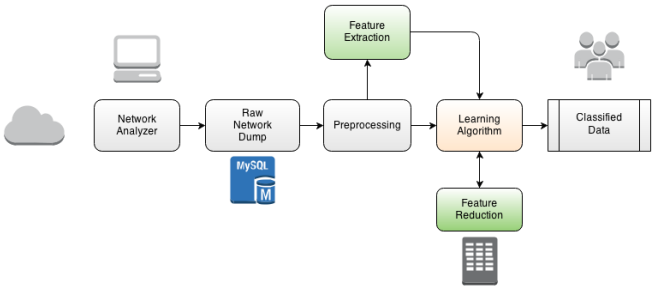

Fig. 1. Proposed network intrusion detection system.

V. Dataset Information

The dataset for our experiment was collected from the 1999 DARPA KDD Cup intrusion detection evaluation program which was prepared and managed by MIT Lincoln Labs. They created one simulation environment similar to U.S. Air Force LAN and injected attack in it. Those attacks fall into four major categories:

- DOS: Denial of Service.
- R2L: Unauthorized access to the local system from a remote host.
- U2R: Unauthorized access to the root of a local system.
- Probe: Sensing network from outside to detect vulnerabilities.

The dataset contains 24 types of attacks in training set and an additional 14 types in testing set. Table I shows different anomaly or attack types in detail.

A. Denial of Service

This type of attack is done by the illegitimate users in such a way that the target system cannot provide efficient service to the legitimate users due to the resource constraint caused by the attackers. For example, back, teardrop, neptune are the DOS attacks.

B. User to Root

Usually these sort of attacks are originated from within the own group. An attacker tries to access the root of the system using a normal user account. For example, perl, loadmodule are well known u2r attacks.

C. Remote to User

Remote users who do not have any account on some specific machines, send packets to explore any vulnerabilities. The prime objective of the attackers is to gain local access to that machine. Dictionary, imap are examples of this type of attack.

D. Probe

Attackers scan a set of networked computers to collect any known or foreseeable vulnerabilities from the network in this sort of attack. An attacker with the configuration and network architecture can easily use this information to look for loopholes.

TABLE I: Anomaly Types

Attack Type	Exploits
DOS	back, land, neptune, pod, smurf, teardrop
U2R	buffer_overflow, load_module, perl, rootkit
R2I	ftp_write, guess_pass, imap, multi hop, phi, spy, warezclient, warezmaster
Probe	Ip_sweep, saint, satan, Nmap

E. Derived Features

The derived features were categorized mainly into three types: basic features, content features, and traffic features. Basic features include individual TCP connections, Content features are suggested within a connection using domain knowledge, and Traffic features are computed using a two-second time window. Table II presents several features, description and type mentioned in the dataset.

TABLE II: Derived Feature Types

Feature Name	Description	Type
duration	length (number of seconds) of the connection	Continuous
protocol type	type of the protocol e.g. tcp, udp, icmp etc.	Discrete
land	1 if connection is from/to the samehost/port; 0 otherwise.	Discrete
urgent	number of urgent packets	Continuous
hot	number of ”hot” indicators	Continuous
su_attempted	1 if ”su root” command attempted; 0 otherwise.	Discrete
count	number of connections to the same host as the current connection in the past two seconds	Continuous

VI. Experimental Results and Evaluation

For the simplicity we have used the data from NSL-KDD which is originally derived from KDD Cup 1999. The original NSL-KDD dataset has several deficiencies, among them one of the most important is the presence of huge number of redundant records. We have solved here using discretization method. This removes the redundancy in such a way that the overall performance indicator provides the accuracy in the result.

Table III shows the classification accuracy among the normal and anomalous data from the test data set which we achieved by taking the subset of the original KDD 1999 dataset [25]. It uses the 20% data of the total set. We have used three different classification algorithms, for example, Naive Bayes, Random Forest, and PART.

TABLE III: EVALUATION ON REDUCED TRAINING DATASET

Naïve Bayes			
	$k = 0$	$k = 1$	$d = 1$
Training	93.57%	95.50%	93.57%
10-fold-cv	93.53%	95.18%	93.53%
Random Forest			
	$t = 5$	$s = 4$	$d = 1$
Training	99.34%	99.36%	99.40%
10-fold-cv	98.70%	98.71%	98.75%
PART			
	$cf = 0.25$	$cf = 1$	$cf = 3$
Training	99.85%	99.90%	99.88%
10-fold-cv	99.43%	99.40%	99.60%

We used MySQL open source database for preprocessing the whole dataset from KDD. We have run different SQL operation to minimize the redundancy from the dataset. We used MATLAB and WEKA 3.7.9 [26] for the classification and feature reduction process. The training phase was performed using both total instance and 10-fold cross validation. Cross validation works well for the large amount of dataset. This process divides the training data into 10 different, disjoint, equal subset and considers 9 sets as training and 1 set as testing. This process is repeated 10 times so that each set can be part of the test set at least once. Validation process ensures that the proposed model works fine for all other instances. Finally, our result shows that the Random Forest, Naive Bayes and PART produces best outcome based on the reduced feature set. Table IV shows detailed accuracy by class for Random Forest. We have also gathered information of our classification using neural network [27] and support vector machine (SVM) [28] for analysis. Table V shows the dataset used in this experiment of both normal and anomaly class. Table VI shows the anomaly class after taking 20% from total 125973 data.

A. Random Forest and Naïve Bayes

This ensemble learning mechanism works in two main phases. Initially, all the possible condition checking for valid outcome is performed using Decision Tree algorithm which is considered as weak in decision making events when the available input features are large in number. Secondly, all those weak outputs are analyzed to get better outcome. In our proposed method, we suggested to use *Random Forest* and *Naive Bayes* as a learning method. We have compared other algorithms in Fig. 2. Some techniques exploit slightly better result but they cannot provide similar result in different dataset. Thus, we considered these two algorithms as ideal for this specific intrusion detection scenario.

B. k-Nearest Neighbor

k-NN works in such a way so that it can classify any unlabeled data with the help of previous labeled training sample. There are different variant of *k*-NN algorithms present in different machine learning tools and IBK is one of the best among them. We used different settings for the data to test the classification accuracy, and IBK showed best accuracy. Although, *k*-NN provides biased result if the training data is not sufficient and the value of *k* does not suffice the requirement of density. If the sample data is very sparse, then lower value of *k* produces wrong classification, thus we have used enough dataset to use lower *k* value (k=1) to increase the accuracy.

TABLE IV: DETAILED ACCURACY BY CLASS: 10-FOLD CROSS VALIDATION FOR RANDOM FOREST

Type	*TP Rate*	*FP Rate*	*Precision*	*Recall*
Normal	0.999	0.002	0.998	0.999
Anomaly	0.998	0.001	0.999	0.998
Type	*F-Measure*	*MCC*	*ROC Area*	*PRC Area*
Normal	0.999	0.998	1.000	1.000
Anomaly	0.999	0.998	1.000	1.000

TABLE V: DATASET USED IN THE EXPERIMENT

Category	*No. of instances*
Normal	67343
Anomaly	58630
Total	125973

TABLE VI: DISTRIBUTION OF REDUCED DATASET

Category	*No. of instances*	*Contribution*
DOS	9234	Continuous
U2R	11	Continuous
R2L	209	Continuous
Probe	2289	Continuous

C. Feature Reduction

In most of the data mining applications we have to deal with large amount of information along with numerous features. It becomes very hectic and complex to generate models that can perform well. So, we need to come up with some solution than can either reduce the number of data samples or the number of attributes/features. This reduction scheme has to be applied before any learning algorithm processes the dataset to have better accuracy and faster computation.

There are two most commonly used method, wrapper and filter for feature reduction. Wrapper evaluates the effectiveness of the feature set, filter method uses heuristics. In this paper we have used Correlation feature selection and Consistency subset method as attribute evaluator along with five different search algorithms (Table VII). Each pair suggests different set of features as the selected attribute, but there are several common attributes in all of the selection. So, we have used the combined set from those outcomes.

1) Correlation feature selection

This feature selection method finds the best possible subset from the large number of feature set so that the subset has high correlation with the class but very low inter correlation among them [29].

$$Merit_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (1)$$

where r_{cf} refers to the average correlation between all the features and class. r_{ff} indicates the average correlation among the features. The CFS feature selection criterion is defined as the below equation which gives the maximum value:

$$CFS = \left\{ \frac{r_{cf1} + r_{cf1} + \ldots + r_{cfk}}{\sqrt{k + 2(r_{f1f2} + \ldots + r_{fif}\, j + \ldots + r_{fkf1})}} \right\} \quad (2)$$

2) Consistency subset

This feature search method evaluates the optimal subset among all the features so that it is the smallest and can identify instances of the individual classes. The below equation was used in evaluating the optimal set when consistency subset was used as attribute selector.

$$Consistency_s = 1 - \frac{\sum_{i=0}^{J} |D_i| - |M_i|}{N} \quad (3)$$

where *s* is an attribute subset, *J* refers to the number of distinct combinations of the attribute values of *s*. $|D_i|$ and $|M_i|$ are the occurrence and cardinality of the *i*-th instance. *N* is the total number of instances or records.

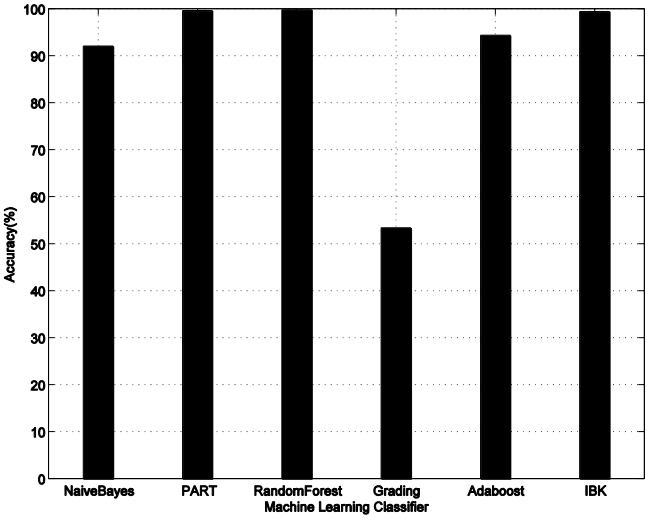

Fig. 2. Classification accuracy for different learning/classification algorithms. The major parameters were tuned for each of the execution.

3) Performance evaluation scheme

The performance measurement for the classifiers was presented in terms of confusion matrix and root mean squared error. Table VIII shows sample confusion matrix for the Random Forest learning algorithm.

Confusion Matrix method evaluates the classification performance in terms of sensitivity or recall. Generally it applies for two class problem but can be extended to further classes as well. For two class problem there can be four different possibilities, true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*). True positive rate is defined as *TPR* = *TP* / *TP* + *FN*. *TPR* is called as Sensitivity or Recall as well. False positive rate is defined as *FPR* = *FP* / *TN* + *FP*. From these two relation, the accuracy of a classifier is defined using the below equation.

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Table VIII represents one of the confusion matrixes. It represents that the correctly classified normal class data = 67308, actual normal class classified as anomaly = 35. Similarly, 117 anomaly class data were classified as normal.

TABLE VII: FEATURES REDUCTION

Attribute Evaluator	Search Method	No. of Selected Attributes	Selected Attributes
CFS	Genetic Search	15	4,5,6,8,10,12,17, 23,26,29,30,32,36 38,39
CFS	PSO Search	9	4,5,6,12,26,29,30, 37,39
CFS	Best First	6	4,5,6,12,26,30
CFS	Evolutionary Search	18	3,4,5,6,8,17,19,23 25,26,29,30,33, 34,37,38,39,41
Consistency Subset	Greedy Stepwise	10	1,3,4,5,14,23,32,3 4,35,37

TABLE VIII: CONFUSION MATRIX FOR RANDOM FOREST

a	*b*	Classified as
67308	35	*a* = normal
117	58513	*b* = anomaly
67425	58548	Total

VII. CONCLUSION AND FUTURE WORK

In the previous sections, we tried to present different scenarios while classifying the dataset using several well known machine learning algorithms. A single learning algorithm can produce significant improvement in classifying when we can adjust some key parameters. We analyzed those details and suggested the best configuration that should be used while solving this particular type of problem.

In our future work, we will use evolutionary algorithms to further accelerate the classification speed and accuracy. Additionally, we have the plan to implement an online NIDS which can provide real-time feedback to the system, so that the unintentional delay from the offline detection method can be eradicated.

REFERENCES

[1] L. Herrero *et al.*, "RT-MOVICAB-IDS: Addressing real-time intrusion detection," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 250-261, 2013.

[2] 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.

[3] M. Tavallaee *et al.* "A detailed analysis of the KDD CUP 99 data set," in *Proc. the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications*, 2009.

[4] G. Kim, S. Lee, and S. Kim. "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4. pp. 1690-1700, 2014.

[5] B. Luo and J. Xia, "A novel intrusion detection system based on feature generation with visualization strategy," *Expert Systems with Applications*, 2014.

[6] C. J. Fung and R. Boutaba, "Design and management of collaborative intrusion detection networks," in *Proc. 2013 IFIP/IEEE International Symposium on Integrated Network Management*, 2013.

[7] K. Alsubhi, M. F. Zhani, and R. Boutaba, "Embedded Markov process based model for performance analysis of Intrusion Detection and Prevention Systems," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2012.

[8] J. W. Haines *et al.*, "Extending the DARPA off-line intrusion detection evaluations," in *Proc. IEEE Conference on DARPA Information Survivability, Exposition II, DISCEX'01*, vol. 1, 2001.

[9] R. Neill and L. P. Carloni, "A scalable architecture for intrusion-detection systems based on a broadband network of embedded set-top boxes," in *Proc. 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2011.

[10] G. V. Kumar and D. Krishna Reddy, "An agent based intrusion detection system for wireless network with artificial immune system (AIS) and negative clone selection," in *Proc. 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies (ICESC)*, 2014.
[11] S. A. Zonouz *et al*., "RRE: A game-theoretic intrusion Response and Recovery Engine," in *Proc. IEEE/IFIP International Conference on Dependable Systems & Networks*, 2009.
[12] W. Hu, J. Gao, Y. Wang, O. Wu, and S. Maybank. "Online adaboost-based parameterized methods for dynamic distributed network intrusion detection," *IEEE Transactions on Cybernetics*, vol. 44, no. 1. pp. 66-82, 2014.
[13] M. Bhuyan, D. Bhattacharyya, and J. Kalita, *Network Anomaly Detection: Methods, Systems and Tools*, pp. 1-34, 2013.
[14] D. Mutz, G. Vigna, and R. Kemmerer, "An experience developing an IDS stimulator for the black-box testing of network intrusion detection systems," in *Proc. 19th Annual Conference on Computer Security Applications*, 2003, pp. 374-383.
[15] F. I. Shiri, B. Shanmugam, and N. B. Idris, "A parallel technique for improving the performance of signature- based network intrusion detection system," in *Proc. 2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011.
[16] A.S Alghamdi *et al*., "Intrusion detection using PCA based modular neural network," *International Journal of Machine Learning and Computing*, vol. 2, no. 5, pp. 583-587, October 2012.
[17] D. K. Bhattacharyy and J. K. Kalita, *Network Anomaly Detection: A Machine Learning Perspective*, CRC Press, 2013.
[18] A. Sundaram, "An introduction to intrusion detection," *Cross-Roads*, vol. 2, no. 4, pp. 3-7, 1996.
[19] L. M. L de Campos, R. C. L. de Oliveira, and M. Roisenberg, "Network intrusion detection system using data mining," *Engineering Applications of Neural Network Springer Berlin Heidelberg*, pp. 104-113, 2012.
[20] K. Wang and S. J. Stolfo, *Anomalous Payload-based Network Intrusion Detection. Recent Advances in Intrusion Detection*, Springer Berlin Heidelberg, 2004.
[21] R. Perdisci *et al*., "McPAD: A multiple classifier system for accurate payload-based anomaly detection," *Computer Networks*, vol. 53, no. 6, pp. 864-881, 2009.
[22] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, 2009.
[23] M. V. Joshi, R. C. Agarwal, and V. Kumar, "Predicting rare classes: Can boosting make any weak learner strong?" in *Proc. the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 297-306.
[24] N. Abe, B. Zadrozny, and J. Langford, "Outlier detection by active learning," in *Proc. the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 504-509.
[25] E. P Guilln, J. R. Parra, and R. V. P. Mendez, "Improving network intrusion detection with extended KDD features," *IAENG Transactions on Engineering Technologies*, Springer Netherlands, pp. 431-445, 2014.
[26] M. Hall, E. Frank, G. Holmes *et al*., "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10-18, 2009.
[27] A. Husagic-Selman, R. Koker, and S. Selman, "Intrusion detection using neural network committee machine," in *Proc. IEEE XXIV International Symposium on Information, Communication and Automation Technologies (ICAT)*, 2013, pp. 1-6.
[28] C. C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, 2011.
[29] L. Yu and H. Liu, "*Feature selection for high-dimensional data: A fast correlation-based filter solution*," in *Proc. the Twentieth International Conference on Machine Learning*, 2003, vol. 3, pp. 856-863.

Md. Enamul Haque is pursuing M.Sc. degree in computer engineering at King Fahd University of Petroleum and Minerals, Saudi Arabia. He completed his B.Sc in computer science and IT from Islamic University of Technology (IUT), OIC. His research interest includes autonomous sensor systems, wireless ad-hoc networks, computer vision and image processing. Previously he worked as a software engineer in Grameenphone IT limited (acquired by Accenture plc.), Bangladesh

Talal Al Kharobi is an assistant professor at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He received his PhD degree in computer engineering from Texas A&M University in 2004. He also received his BSc and MSc degree in computer engineering from King Fahd University of Petroleum and Minerals in 1993 and 1997 respectively. His research interests include ANN, VLSI, and information security.