I. J. Computer Network and Information Security, **2015, 3, 35-41**
Published Online February 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2015.03.05

Effective Hybrid Intrusion Detection System: A Layered Approach

Abebe Tesfahun, D. Lalitha Bhaskari
AUCE (A), Andhra University, Visakhapatnam, AP, India
Email: abesummit@yahoo.com, lalithabhaskari@yahoo.co.in

Abstract—Although there are different techniques proposed for intrusion detection in the literature, most of them consider standalone misuse or anomaly intrusion detection systems. However, by taking the advantages of both systems a better hybrid intrusion detection system can be developed. In this paper, we present an effective hybrid layered intrusion detection system for detecting both previously known and zero-day attacks. In particular, a two layer system that combines misuse and anomaly intrusion detection systems is proposed. The first layer consists of misuse detector which can detect and block known attacks and the second layer comprises of anomaly detector which can efficiently detect and block previously unknown attacks. The misuse detector is modeled based on random forests classifier and the anomaly detector is built using bagging technique with ensemble of one-class support vector machine classifiers. Data pre-processing is done using automatic feature selection and data normalization. Experimental results show that the proposed intrusion detection system outperforms other well-known intrusion detection systems in detecting both previously known and zero-day attacks.

Index Terms—Intrusion, Hybrid, Misuse, Anomaly, Random Forests, Performance.

I. Introduction

The information communication infrastructure has highly improved the lives of modern society. However, this infrastructure is always under the threats of intrusion and misuse. In order to prevent such threats the research and industry community have come up with different threat detection and prevention technologies. One of such technologies is Intrusion Detection Systems (IDS). An intrusion detection system monitors and analyzes the events occurring in a computer system or network environment and alerts a human operator to the presence of possible incidents that violate standard security practices [1]. Based on the deployment area intrusion detection technologies could be categorized as Host-based IDS (deployed at individual computers) or Network-based IDS (deployed at network level). According to the methods used for analyzing the collected data, IDS can also be categorized into two broad categories: Misuse based detection and anomaly based detection.

Misuse based (signature based) intrusion detection system tries to detect malicious activities based on patterns or signatures of known attacks. If a pattern match is detected, an alarm is reported to the network administrator. Since misuse based detection system is specifically designed for detecting known attacks, it generates low number of false alarms. However, misuse based intrusion detection systems could not detect novel attacks [2].

Anomaly based intrusion detection refers to identifying events that are anomalous with respect to the normal system behavior. If the incoming network traffic patterns do not follow the normal network traffic behavior, an alarm will be reported and such patterns are called anomalies or outliers [2]. Despite their capability in detecting novel attacks anomaly based intrusion detection systems suffer from high false positive rate.

The misuse and anomaly based intrusion detection systems are complementary [3]. Hence, by taking advantages of low false-positive rate by signature-based intrusion detection system and the ability of anomaly detection system to detect zero-day attacks some researchers introduce a hybrid intrusion detection system. According to the fusion mechanisms of the two systems, hybrid intrusion detection systems can have layered or parallel architecture.

M. A. Aydın et al. [4] proposed a hybrid IDS by sequentially combining packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD) with Snort. They used the anomaly detectors (PHAD and NETAD) as a preprocessor for Snort.

On the other hand Depren et al. [5] proposed a parallel hybrid IDS architecture using Self-Organizing Map as anomaly detector and J.48 decision tree for misuse detection. In their proposed framework decision support system is implemented for combining the detection results of the two systems.

Though, many intrusion detection frameworks and systems have been developed by the research community, IDS performance and zero-day attack detection are still open research issues and challenges. Hence, by taking the advantage of low false alarm of misuse based intrusion detection system and the capability of detecting zero-day attacks by anomaly based intrusion detection system in this paper we proposed a hybrid layered intrusion detection system. We used random forests classifier for misuse detection and ensemble of one-class support vector machine (1-SVM) with bagging for anomaly

detection. In the proposed system automatic feature selection and data normalization are used for data pre-processing. The misuse detection model is built using a training data containing normal profiles and known attacks and for building the anomaly detector model we used only normal network traffic training data.

The subsequent sections of this paper are organized as follows. In section II we present some theoretical background information about random forests, one-class SVM, and data pre-processing techniques. Section III describes the proposed hybrid intrusion detection system. The dataset used for the experiment and performance evaluation results will be discussed in section IV. Finally, we will conclude this research work in section V.

II. Theoretical Background

A. Random Forests

As the developer of random forests classifier, L. Breiman, defined random forests as ensemble of many decision trees and each tree is developed using a bootstrapped sample from the original training data [6]. If the total number of attributes is M, then for each tree only m attributes are chosen randomly (where m < M). For each tree randomness is introduced in two ways: during bootstrapped sample generation and random candidate feature selection at each node of a tree. Once the classifier is built, to classify an incoming test data, the input vector of incoming data will be put down to each of the trees in the forest. The majority vote from predictions of the ensemble of trees is used to decide to which class the instance under consideration belongs. Compared to a conventional decision trees random forests classifier has better detection accuracy [7, 13].

In the proposed system, the random forest classifier is used for misuse detection. A dataset containing both normal and known attacks is used to train the random forests classifier.

B. Feature Selection

One of the major issues associated with large dataset, like network intrusion detection dataset, is dimensionality problem. In the case of large dataset feature selection is a crucial step for dimensionality reduction. Feature selection is a process of data dimensionality reduction by determining whether a feature is relevant or not for a given problem. The target of feature selection is to select feature vector that leads to large between-class distance and small within-class variance [8]. Using effective features in designing a classifier not only can reduce the data size but also can improve the performance of the classifier and enhances data understanding and visualization [8].

Generally, there are three types of feature selection models: Filter, Wrapper and Hybrid [15]. In wrapper model the feature selection is dependent on the performance of the learning algorithm. On the other hand, in filter model the quality of selected features is dependent only on the statistical property of the data.

In this paper filter type of feature selection algorithm is implemented using Information Gain (IG) of an attribute. As stated in [7] the information gain for a given input feature X with respect to the class attribute C can be computed as follows:

$$I(C;X) = H(C) - H(C \mid X) \tag{1}$$

Where

$$H(C) = -\sum_{i=1}^{n} P(C = c_i) \log_2 P(C = C_i) \tag{2}$$

$P(C = c_i)$ is the probability that the class attribute c_i occurs, and

$$H(C \mid X) = -\sum_{i=1}^{m} P(X = c_i) H(C \mid X = c_i) \tag{3}$$

H(C) is entropy of C and $H(C \mid X)$ is the average conditional entropy of C. In this paper, X defines individual input attributes in the training dataset, and C defines class label (Normal or Attack).

C. One Class Support Vector Machine

A Support Vector Machine (SVM) is a learning systems based on mapping the training data into a high dimensional feature space using some non-linear mapping functions. By projecting to a high dimensional feature space a non-linear decision boundary can be built. Scholkopf et al. [10] motivated by SVM proposed one-class SVM classification. The idea behind one-class SVM is to determine a hyperplane that separates the required fraction of one class training patterns from the origin in the feature space F. In one-class SVM there is a tradeoff between maximizing the distance between the origin and the separating plane and a rejection rate. It is not always the case to find the separating hyperplane in the original feature space. Hence, the function Φ: X→F is used for mapping the original feature space to kernel space.

The objective function for one-class SVM is formulated using quadratic programming minimization as follows

$$\min_{W,\xi_i,\rho} \left(\tfrac{1}{2}\|W\|^2 - \rho + \tfrac{1}{\nu n}\sum_{i=1}^{n} \xi_i \right) \tag{4}$$

Subjected to:

$$W.\phi(x_i) \geq \rho - \xi_i, \quad \xi_i \geq 0 \quad \forall i = 1,2,...,n$$

Where x_i is the i-th training instance, n is the number of training instances, ρ is the margin, **w** is a normal vector to the hyperplane, ν represents fraction of outliers. For each training instance i, there is a slack variable ξ_i associated with a penalty for rejection.

If the minimization problem in equation 4 solved using Lagrange multipliers in quadratic programing, the decision function only depends on the dot-product of the vectors in the feature space. Hence it is not necessary to

perform an explicit mapping to that space rather the dot product in the feature space can be computed using kernel function $K(x)=\phi(x)^T\phi(x)$. Using the Lagrange multipliers and kernel trick the decision function for test data z will be as follows:

$$F(z)=\text{sgn}((\sum_{i=1}^{n}\alpha_i K(z,x_i))-\rho) \quad (5)$$

α_i is a Lagrange multiplier. Since majority of α_i are zero, the computation of F(z) in equation 5 is efficient. If $F(z)\geq 0$, then the test data z is similar to the training data and classified to the class of the training class. Otherwise, the test data will be rejected as an outlier. $K(z,x_i)$ is a kernel function. There are different types of kernel function such as linear, polynomial, Gaussian radial basis, and sigmoid. However, Gaussian radial basis function (RBF) shown in equation 6 is the most commonly used function for anomaly detection.

$$K(z,x_i)=e^{-\gamma\|z-x_i\|^2} \quad (6)$$

The value of γ determines how much a support vector influences its neighbors. If γ is large, most of the training vectors will be support vectors. There will be few support vectors if the value of γ is small.

D. Normalization

In publicly available intrusion detection dataset some of the features are nominal and others are not normalized. For learners that learn from statistical characteristics of features normalization of data is a crucial pre-processing step. Data normalization is the way of scaling the values of each attribute to fall within a specific range so that the effect of one attribute should not dominate the others. Feature based data normalization can be broadly categorized into linear and non-linear methods. For our system we used min-max based linear data normalization technique. The formula for min-max based normalization is shown in equation 7.

$$X' = \frac{X-\min_A}{\max_A-\min_A} \quad (7)$$

X and X' are value to be normalized and the normalized attribute value respectively. $\min_A$ and $\max_A$ are the minimum and maximum possible values for attribute A before normalization.

For handling nominal features for SVM, it is important to convert them to numeric representation. Hsu [11] recommended the conversion of nominal features in to binary representation prior to using them for SVM.

III. Proposed Hybrid Intrusion Detection System

The proposed hybrid layered intrusion detection framework is shown in Fig. 1. The information gain (IG) based feature selection module computes the information gain of each attribute using equation 1. Once the information gain of each attribute is computed the next step is to select optimal subset of features for the classifier. For optimal feature selection we used Algorthim-1.

In Algorthim-1 X is the original feature vector and IG is information gain for the respective features. The selected feature vector is represented by Y. T is some threshold value that used for selecting optimal features subset. T is dependent on the training data used for classification.

Algorithm-1: Optimal Feature Subset Selection

Input: $X=\{X_1, X_2,\ldots,X_m\}$, $IG=\{IG_1, IG_2,\ldots, IG_m\}$ and T
Output: $Y=\{Y_1, Y_2,\ldots,Y_n\}$

1: $IGTotal \leftarrow \sum_{i=1}^{m} IG_i$
2: $X' \leftarrow Descending\ Sort(X, IG)$
3: **For** i is from 1 to m **do**
4: $Wx'_i \leftarrow \frac{IG_i}{IGTotal}$
5: **End for**
6: $S \leftarrow 0$
7: **For** i is from 1 to m and $S \leq T$ **do**
8: $S \leftarrow S + Wx'_i$
9: $n \leftarrow i$
10: $Y_i \leftarrow X'_i$
11: **End for**

As soon as the relevant features are identified, then the feature selection module will forward the data with the selected features to the misuse intrusion detector. The misuse detector is implemented using random forests classifier. The random forests classifier model is built using training data which contains both normal and known attack patterns. The random forests classifier approach provides mechanisms for excluding well-known attacks from being reprocessed by the subsequent one-class SVM based anomaly detector. Only patterns which are classified as normal by the random forests classifier are forwarded to anomaly detector module for final decision.

The random forests classifier-based misuse detector is autonomous. It can block the detected attacks without waiting for the decision of the anomaly detector. Such capability of the misuse detector somehow reduces the time required to detect known attacks. If unlabeled network traffic instances arrived to the proposed model, then those instances classified as attack by the random forests classifier would be blocked whereas those instances classified as normal by the random forests classifier would be sent to the next level.

Training data, which are classified as normal by the random forests classifier using cross validation, are used for feature selection for the one-class SVM classifier. Though the data classified as normal by random forests classifier is imbalanced (large number of normal profiles and small number of attacks), it is enough for selecting relevant features for identifying anomalies from normal profiles.

The next component, data normalization module, will preprocess the incoming data to enhance the performance of one-class SVM. Normalization is made using equation 7. In order to handle nominal values we converted all nominal attributes in to binary representation as recommended by Hsu [11].

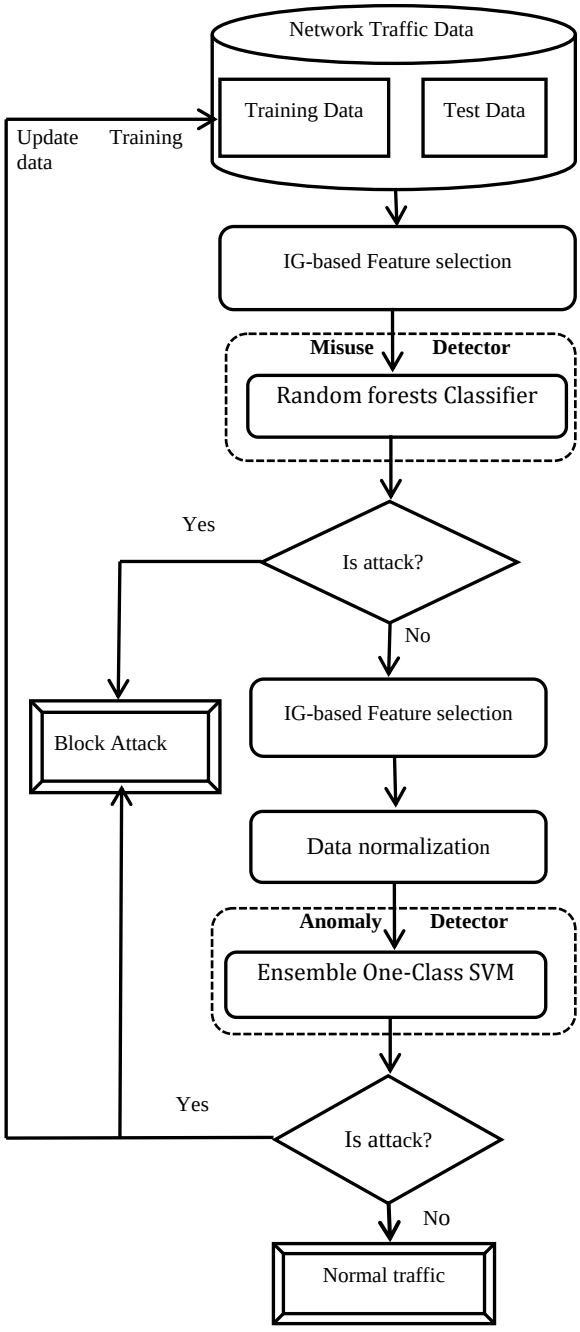

Fig. 1. Proposed Hybrid Layered Intrusion Detection System.

The anomaly detector is built using only normal training instances which are also classified by the misuse detector as normal. Instead of single one-class SVM, to improve the performance of the anomaly detector, we proposed the use of ensemble of one-class SVM classifiers with bootstrap aggregating technique. As it has been done in random forests algorithm, each of the one-class SVM classifiers is built using a bootstrapped sample of training instances. The decision of each of these classifiers is aggregated and a majority voting is done for final decision. The anomaly detector will block detected attacks which were considered as normal traffic by the misuse detector. The signatures of these attacks will be used for updating the original training data.

IV. Experiments

All the experiments in this paper are implemented using WEKA 3.7.9 machine learning tool and MATLAB 2013a.

A. Dataset Description

The proposed system is evaluated using publicly available NSL-KDD intrusion detection dataset which is an enhanced version of the KDD99 intrusion detection dataset. KDD99 dataset is the only well-known and publicly available data set in the area of intrusion detection [14]. It is still widely used in evaluating the performance of proposed intrusion detection algorithms. On the KDD99 intrusion detection dataset 78% of training instances and 75% of test instances are duplicated. Hence the NSL-KDD dataset is generated by removing redundant instances in both the training and test data of the KDD99 intrusion detection dataset [12]. This dataset has 41 features and one class attribute. The training data contains 24 types of attacks and the testing data contains extra 14 types of attacks. The attacks in this dataset are categorized in one of the four attack categories (DoS, Probing, User to Root and Remote to Local attacks)

Though NSL-KDD dataset is enhanced version of the KDD99 dataset we observed two basic problems in this dataset. First as shown in Fig. 2 there are ambiguities in some records of the testing dataset. That is some records have same value for all the 41 features, however they are labeled to different classes (one as normal and the other as attack). The second observation we made is there is a feature called *num_outbounds_cmds* which has a value of zero for all the records in both the training and testing data. This feature will not have any contribution in identifying attacks from normal profiles. Hence we made two improvements in using NSL-KDD dataset: we removed all ambiguous records and the *num_outbounds_cmds* feature from the dataset. The distribution of the dataset used in this experiment is depicted in Table 1.

B. Data Pre-processing

After calculating information gain for each of the features in the training data, for the random forests classifier we selected 20 features by applying the optimal feature selection algorithm with T=0.9. List of all the selected features with their information gain value (IG) is shown in Table 2. The random forests classifier is built using the selected features and the KDDTrain+ full training data. For the ensemble one-class SVM classifiers 11 features are selected from the 20 features by applying optimal feature selection algorithm with T=0.9

on the portion of training data which are classified as normal by the random forests classifier. The bootstrap training sample for each of the 1-SVM classifier is generated from 20% of the normal KDDTrain+ instances which are also classified as normal by the random forests classifier.

Record#	
7833	0,udp,private,SF,105,105,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,255,255,1,0,0,0,0,0,0,0,**normal**
8327	0,udp,private,SF,105,105,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,255,255,1,0,0,0,0,0,0,0,**snmpgetattack**
11695	0,udp,private,SF,105,105,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,3,0,0,0,0,1,0,0,255,255,1,0,0.01,0,0,0,0,0,**normal**
16303	0,udp,private,SF,105,105,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,3,0,0,0,0,1,0,0,255,255,1,0,0.01,0,0,0,0,0,**snmpgetattack**

Fig. 2. Sample ambiguous records in the NSL-KDD Test dataset

Table 1. Experiment dataset distribution

Data	Normal	Attack
Training data	67,343	58,630
Test data	9,515	12,663

Table 2. Selected features list

Rank	Feature name	IG
1	srs_bytes	0.816
2	Service	0.672
3	dst_bytes	0.633
4	Flag	0.519
5	diff_srv_rate	0.519
6	same_srv_rate	0.510
7	Dst_host_srv_count	0.476
8	Dst_host_same_srv_rate	0.438
9	Dst_host_diff_srv_rate	0.411
10	Dst_host_serror_rate	0.406
11	logged_in	0.405
12	Dst_host_srv_serror_rate	0.398
13	serror_rate	0.393
14	Count	0.384
15	srv_serror_rate	0.379
16	Dst_host_srv_diff_host_rate	0.271
17	Dst_host_count	0.198
18	Dst_host_same_src_port_rate	0.189
19	srv_diff_host_rate	0.142
20	srv_count	0.094

C. Performance Measure

In this paper attack detection rate (DR), false positive rate (FPR) and receiver operating characteristic (ROC) curve are used for performance evaluation and comparison of classifiers. In the case of intrusion detection, detection rate is the ratio of the number correctly detected attacks to the total number of attacks. False positive rate is the ratio of the number of normal traffic data which are detected as attack to the total number of normal data. The ROC curve is used for comparing the proposed hybrid system with the individual standalone random forest based misuse detector and anomaly based SVM binary classifier. Depren et al. [5] evaluated their proposed system using cross validation with the KDD99 training dataset. However, such way of evaluation in intrusion detection doesn't include novel attacks in the test data. For evaluating our system we used the test data which contains 14 more attack types than the training data.

The performance of the proposed hybrid layered intrusion detection system for the test data with different attack distributions is shown in Table 3. For these specific results Gaussian Basis kernel Function (RBF) with parameter $\gamma = 0.02$ and $\nu = 0.1$ were used for one-class SVM learner. As it can be visualized from the ROC curve shown in Fig. 3 the proposed system outperforms standalone random forests-based misuse detector and ordinary SVM-based anomaly detector. This ROC curve is generated for 50% of the attacks in the NSL-KDDTest+ dataset.

Attack detection rate and false positive rate for the proposed system and some well-known models using full test data is shown in Table 4. Attack detection rate and false positive rate for the proposed system is 92.13% and 6.42% respectively. The result shows that compared to some well-known methods the proposed system can effectively detect attacks.

Table 3. Performance of the proposed hybrid intrusion detection system for the test data

Attack distribution in test data	Detection rate	False Positive rate
1% Attack	0.8968	0.0642
2% Attack	0.9091	0.0642
5% Attack	0.9303	0.0642
10% Attack	0.9202	0.0642
20% Attack	0.9194	0.0642
30% Attack	0.9205	0.0642
50% Attack	0.9253	0.0642
100% Attack	0.9213	0.0642

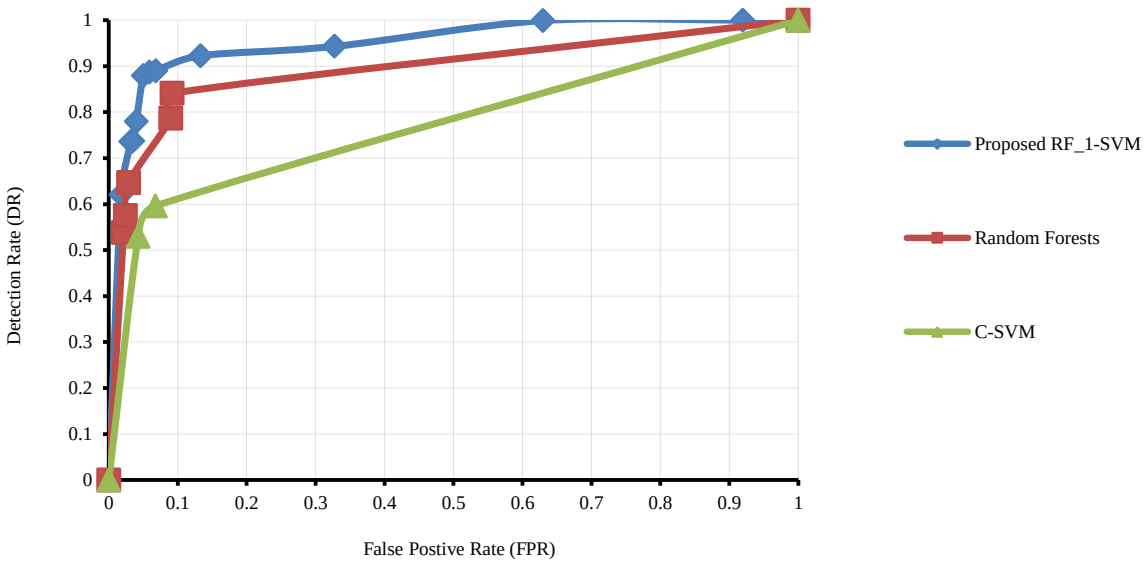

Fig. 3. ROC curve for the test data with 50% attack distribution when γ= 0.01

Table 4. Performance comparison for the full test data

Model	Attack Detection Rate (DR)	False Positive Rate (FPR)
Naive Bayes	0.588	0.063
J48 decision tree	0.647	0.029
Random forests	0.734	0.031
SVM(γ = 0.02)	0.591	0.068
Proposed RF_1-SVM (γ=0.02 and ν =0.1)	0.9213	0.0642

V. Conclusion

In this paper a hybrid layered IDS was presented by combining both misused and anomaly intrusion detection systems. The random forests classifier was used to detect previously known attacks. The anomaly detector, ensemble of one-class SVM classifiers, was built using bagging technique. The proposed system addresses the problem of attack detection for previously known and unknown attacks. The experimental results show that the proposed system is very effective in improving attack detection rate with small false positive rate. We compared our approach with some well-known methods and found that the proposed system can effectively detect previously unknown attacks with a detection rate improvement of 18.73%.

In using publically available intrusion detection datasets for training and evaluation of a proposed model care has to be taken. Those ambiguous records in the dataset should be removed. The optimal feature subset selection method presented in this paper has not only contributed for dimensionality reduction it also has contribution for performance improvement for both the misuse and anomaly intrusion detection systems.

Though the developed system updates the training data with new attack types, it is not adaptive. Hence, in our future work, we plan to make the proposed system real time and adaptive to cope with dynamic attack scenarios.

References

[1] Scarfone, K., Mell, P., Guide to Intrusion Detection and Prevention Systems (IDPS). *NIST Special Publication* 800-94, 2007.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *Communications Surveys & Tutorials*, IEEE press, vol. 16, no. 1, pp. 303 – 336, 2013.

[3] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion Detection System: A Comprehensive Review," *Journal of Network and Computer Applications*, vol. 36, Issue 1, pp. 16-24, 2013.

[4] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A Hybrid Intrusion Detection System Design for Computer Network Security," *Computers and Electrical Engineering*, vol. 35, no. 3, pp.517-526, 2009.

[5] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks," *Expert Systems with Applications*, pp. 713–722, 2005.

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

[7] A. Tesfahun, and D. L. Bhaskari, "Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction*," in Proc. of 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, pp.127-132, 2013.

[8] S. Theodoridis, and K. Koutroumbas, "Pattern Recognition", Academic press, 2009.

[9] B. Scholkopf , R. Williamson, A. Smola , J. Shawe-Taylor, and J. Platt, "Support Vector Method for Novelty Detection," *NIPS*, vol. 12, pp. 582-588, 1999.

[10] Z. Xue-qin, G. Chun-hua, and L. Jia-jin, "Intrusion Detection System Based on Feature Selection and Support Vector machine", *in Proc. of First International Conference on Communications and Networking in China*, pp. 1-5, Oct. 2006.
[11] Hsu, Chih-Wei, Chang, Chih-Chung, and Chih-Jen, "A Practical Guide to Support Vector Classification", National Taiwan University, 2003.
[12] Tavallaee, E. Bagheri, W. Lu, and A.A. Ghorbani "A Detailed Analysis of the KDD CUP 99 Data Set", *in proc. of IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1-6, 2009.
[13] Zhang and M. Zulkernine, "Network Intrusion Detection using Random Forests", School of Computing Queen's University, Kingston Ontario, 2006.
[14] W. Lee, S. Stolfo, P. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, and J. Zhang, "Real Time Data Mining-based Intrusion Detection", *The 2001 DARPA Information Survivability Conference and Exposition (DISCEX II)*, Anaheim, CA, June 2001.
[15] Y. Chen, Y. Li, X. Q. Cheng, and L. Guo, "Survey and taxonomy of feature selection algorithms in intrusion detection system*," in Proc. of the 2nd SKLOIS conference on Information Security and Cryptology*, pp. 153–167, 2006.

Authors' Profiles

Abebe Tesfahun received his B.Sc. degree in Electrical and Computer Engineering and M.Tech in Electronics and Computer Engineering from Addis Ababa University, Ethiopia. He is currently a PhD candidate in Andhra University, Visakhapatnam, India.

His research interest includes Network Security, Critical Infrastructure Protection, and Machine learning.

D. Lalitha Bhaskari is a Professor in the department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India. She is guiding more than 15 PhD Scholars from various institutes. Her main research interest includes Network Security, Image Processing, Pattern Recognition, Steganography and Digital Watermarking.

Prof. D. Lalitha Bhaskari is a member of IEEE, IJSCI, CSI and Associate Member of Institute of Engineers.

