# Layering Based Network Intrusion Detection System to Enhance Network Attacks Detection

**Yi, Mon Aye[1], Phyu, Thandar[2]**

[1]University of Computer Studies, Mandalay, Myanmar

[2]University of Computer Studies, Yangon, Myanmar

**Abstract:** ***Due to continuous growth of the Internet technology, it needs to establish security mechanism. Intrusion Detection System (IDS) is increasingly becoming a crucial component for computer and network security systems. Most of the existing intrusion detection techniques emphasize on building intrusion detection model based on all features provided. Some of these features are irrelevant or redundant. This paper is proposed to identify important input features in building IDS that is computationally efficient and effective. In this paper, we identify important attributes for each attack type by analyzing the detection rate. We input the specific attributes for each attack types to classify using Naïve Bayes, and Random Forest. We perform our experiments on NSL-KDD intrusion detection dataset, which consists of selected records of the complete KDD Cup 1999 intrusion detection dataset.***

**Keywords:** security mechanism, Intrusion Detection System, Naïve Bayes, Random Forest**.**

## 1. Introduction

The field of computer security has become a very important issue for computer systems with the rapid growth of computer network and other transaction systems over the Internet. An Intrusion Detection System (IDS) is a system for detecting intrusions that attempting to misuse the data or computing resources of a computer system. IDS play a vital role in network security. IDS is security tools that collect information from a variety of network sources, and analyze the information for signs of network intrusions.

Depending on the information source considered IDS may be either host or network based. A host based IDS analyzes events such as process identifiers and system calls, mainly related to OS information. On the other hand, a network based IDS analyzes network related events: traffic volume, IP addresses, service ports, protocol usage, etc. This paper focuses on the latter type of IDS.

Depending on the type of analysis carried out intrusion detection systems are classified as either signature-based or anomaly-based. Signature-based schemes (also denoted as misuse-based) seek defined patterns, or signatures, within the analyzed data. For this purpose, a signature database corresponding to known attacks is specified a priori. On the other hand, anomaly-based detectors attempt to estimate the ‘‘normal’’ behavior of the system to be protected, and generate an anomaly alarm whenever the deviation between a given observation at an instant and the normal behavior exceeds a predefined threshold. Another possibility is to model the ‘‘abnormal’’ behavior of the system and to raise an alarm when the difference between the observed behavior and the expected one falls below a given limit.

Signature-based schemes provide very good detection results for specified, well-known attacks. However, they are not capable of detecting new, unfamiliar intrusions, even if they are built as minimum variants of already known attacks. On the contrary, the main benefit of anomaly-based detection techniques is their potential to detect previously unseen intrusion events. However, the rate of false positives in anomaly-based systems is usually higher than in signature-based ones [6].

There have been many techniques used for machine learning applications to tackle the problem of feature selection for intrusion detection. In [23], author used Backward Sequential Elimination (BSE) to reduce features set. In [10], author used gain ration for attribute selection.

We construct system which not only reducing time consuming for features set and but also improving overall accuracy. In this paper we use original NSL-KDD [18] training and the test dataset, which have totally different distributions due to novel intrusions, introduces in the test data. The training dataset is made up of 22 different attacks out of 39 present in the test data. The attacks that have any occurrences in the training sets should be considered as known attacks and others those are absent in the training set and present in the test set, considered as novel attacks.

The rest of this paper is organized as follows. In section 2, we discuss the related work. In section 3, we describe Naïve Bayes and Random Forest. Section 4 presents the Feature Selection. The proposed system is described in section 5. The experiments are presented in section 6. Finally, we conclude the paper in section 7.

## 2. Related Work

Intrusion Detection System (IDS) has increasingly become a crucial issue for computer and network systems. Recently machine learning-based IDSs have been subjected to extensive researches because they can detect both misuse and anomaly. Multi-layer intrusion detection model was proposed in [10]. They used gain ratio for selecting the best features for each layer and classified the system by using machine learning algorithms such as C5.0, Multi-Layer Perceptron (MLP) Neural Networks and Naïve Bayes. In [23] authors proposed a three-layer approach to enhance the perception of intrusion detection on reduced feature set to detect both known and novel attacks. They used NSL-KDD dataset for their experiments. They employed domain knowledge and the Backward Sequential Elimination (BSE) to identify the important set of features and Naïve Bayes classifier for classification. In [13], authors used Naïve Bayesian for classification. Their experiment is implemented on NSL-KDD dataset. The NIDS based on AdaBoost algorithm had designed in [24] using Java Technology and tested it on the NSL-KDD intrusion detection dataset. They had taken 20% of the training dataset of NSL-KDD as input to the system. In [19], author applied Rough set theory for extracting relevant features and Adaboost algorithm for detecting intrusions. Data mining methods, such as decision tree (DT) [15] [7], naïve Bayesian classifier (NB)[13], neural network (NN), Rough Set[5], support vector machine (SVM)[16], k-nearest neighbors (KNN)[5], fuzzy logic model, and genetic algorithm have been widely used to analyze network logs to gain intrusion related knowledge to improve the performance of IDS in last decades.

## 3. Theory Background

### 3.1 Naïve Bayes

Naïve Bayesian classification is called naïve because it assumes class conditional independence. That is, the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is made to reduce computational costs, and hence is considered naïve. The major idea behind naïve Bayesian classification is to try and classify data by maximizing $P(X_j|C_i)|P(C_i)$ (where $i$ is an index of the class) using the Bayes theorem of posterior probability[17]. In general,

- We are given a set of unknown data tuples, where each tuple is represented by an n-dimensional vector, $X = (x_1, x_2,\ldots,x_n)$ depicting n measurements made on the tuple from n attributes, respectively, $A_1,A_2,\ldots,A_n$. We are also given a set of m classes $C_1$, C2… Cm.
- Using Bayes theorem, the naïve Bayesian classifier calculates the posterior probability of each class conditioned on X, X is assigned the class label of the class with the maximum posterior probability conditioned on X. Therefore, we try to maximize $P (X_{ij}|X) = P (X_j|C_i) P (Ci) = P(X)$. However, since P(X) is constant for all classes, only $P (X_j|C_i) P (C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, i,e, $P(C_1)=P(C_2)=\ldots=P(C_m)$, and we would therefore maximize $P(X_j|C_i)$. Otherwise, we maximize $P (X_j|C_i) P (C_i)$. The class prior probabilities may be estimated by $P(C_i)=S_i/S$, where $S_i$ is the number of training tuples of class $C_i$, and s is the total number of training tuples.
- In order to reduce computation in evaluating $P P(X_j|C_i)$, the naïve assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple, i.e., that there are no dependence relationships among the attributes.
- If $A_k$ is a categorical attribute then $P(A_k=x_{kj}|C_i)$ is equal to the number of training tuples in $C_i$ that have $x_k$ as the value for that attribute, divided by the total number of training tuples in $C_i$.
- If $A_k$ is a continuous attribute then $P(A_k=x_{kj}|C_i)$ can be calculated using a Gaussian density function with a mean μ and standard deviation σ defined by

  $g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp - \frac{(x-\mu)^2}{2\sigma^2}$

  So that $p\ (x_k|C_i) = g\ (x_{k,},\mu_{C_i},\sigma_{C_i})$

We need to compute $\mu_{C_i}$ *and* $\sigma_{C_i}$, which are the mean and standard deviation of values of attribute $A_k$ for training samples of class $C_i$.

### 3.2 Random Forest

The random forests [14] are an ensemble of unpruned classification or regression trees. In general, random forest generates many classification trees and a tree classification algorithm is used to construct a tree with different bootstrap sample from original data using a tree classification algorithm. After the forest is formed, a new object that needs to be classified is put down each of the tree in the forest for classification. Each tree gives a vote about the class of the object. The forest chooses the class with the most votes. RF algorithm is given below [4]:

1. Build bootstrapped sample B*i* from the original dataset D, where |B*i*| = |D| and examples are chosen at random with replacement from D.
2. Construct a tree $\tau_i$, using B*i* as the training dataset using the standard decision tree algorithm with the following modifications:
3. At each node in the tree, restrict the set of candidate attributes to a randomly selected subset (*x1, x2, x3, … , xk*), where *k = no. of features.*
4. Do not prune the tree.
5. Repeat steps (1) and (2) for *i = 1, … , no. of trees*, creating a forest of trees $\tau_i$ , derived from different bootstrap samples.
6. When classifying an example x, aggregate the decisions (votes) over all trees $\tau_i$ in the forest. If $\tau_i (x)$ is the class of x as determined by tree $\tau_i$, then the predicted class of x is the class that occurs most often in the ensemble, i.e. the class with the majority votes.

Random Forest has been applied in various domains such as modeling [11] [20], prediction [6] and intrusion detection system [12] [22]. Zhang and Zulkernine [12] implemented RF in their hybrid IDS to detect known intrusion. They used the outlier detection provided by RF to detect unknown intrusion. Its ability to produce low classification error and to provide feature ranking has attracted Lee et al. [22] to use the technique to develop lightweight IDS, which focused on single attack.

## 4. Feature Selection

The 41 features for network connection records fall into three categories [23].

- **Intrinsic features**. Intrinsic features describe the basic information of connections, such as the duration, service, source and destination host, port, and flag.

- **Traffic features.** These features are based on statistics, such as number of connections to the same host as the current connection within a time window.
- **Content features.** These features are constructed from the payload of traffic packets instead of packet headers, such as number of failed logins, whether logged in as root, and number of accesses to control files.

Feature selection is an effective and an essential step in successful high dimensionality data mining applications. It is often an essential data processing step prior to applying a learning algorithm. Reduction of the irrelevant features leads to a better understandable model, enhances the accuracy of detection while speeding up the computation and simplifies the usage of different visualization technique. Thus reducing attribute space improves the overall performance of IDS.

## 5. Proposed System

The proposed system applies data mining techniques to build patterns for network intrusion detection. We implement the layering based approach by selecting small set of features for every layer rather than using all the 41 features. According to serious level we arranged layer 1 is DoS layer, layer 2 is R2L layer, layer 3 U2R layer and layer 4 is Probe layer.

### 5.1 Knowledge Base Attributes Selection

Feature selection is the most critical step in building intrusion detection models [1], [2], [3]. In order to make IDS more efficient, reducing the dimensions and data complexity have been used as simplifying features. Feature selection can reduce both the data and the computational complexity. It can also get more efficient and find out the useful feature subsets [8]. It is the process of choosing a subset of original features so that the feature space is optimally reduced to evaluation criterion. We analyze features by using different machine learning attribute selection algorithm such as ChiSquare Attribute Selection, Cfs Subset Evaluation, GainRation Feature Selection to get optimal features. We compare these features with relevant and meaningful definition of each attack types. The algorithm that provides the reduction of features has been proposed in the following steps:

Step 1: Input NSL-KDD Intrusion Detection Dataset.
Step 2: Eliminate the dispensable attributes.
Step 3: Choose common features from feature selection algorithms.
Step 4: Select relevant and meaningful features, and merge possible features by comparing common features of feature selection algorithm..
Step 5: Compute the detection Rate (DR) of the selected features by choosing different combination of features.
Step 6: Analyze and estimate DR of selected features based on each attack type by using Naïve Bayes and Random Forest.
Step 7: If (DR) is less than Threshold percentage value, go to step 3 and continues.
Step 8: If (DR) is greater than Threshold value, these feature is selected.
Step 9: Assesses the optimal selected features by evaluating the performance value resulted from Naïve Bayes and Random Forest Classifiers.
Step 10: Prove that the optimal selected features with the other classifier such as J48 decision tree, Support Vector Machine, Random Tree.

We select features for each layer based on the type of attacks and shows in Table 1. The selected feature set of proposed model for all layers are:

**Table 1:** Depicts Proposed Feature Set for each Layer

| *Layer Number* | *Attack Type* | *Feature Number* |
|---|---|---|
| layer 1 | DoS | 1,5,23,28,34,39,41 |
| layer 2 | R2L | 3,11,12,21,22,23,32,33 |
| layer 3 | U2R | 3,6,10,13,23,24,35,36 |
| layer 4 | Probe | 1,3,6,12,30,32,35,36 |

We use NSL-KDD dataset for our experiment. This dataset has many advantages over original KDD 1999 dataset. To computer performance of the classification algorithm for each of these feature sets, we used Weka (3.7) machine learning tool [9].

## 6. Experiments

### 6.1 Networking Attack in Data Set

The simulated attacks were classified, according to the actions and goals of the attacker. We use the NSL-KDD [18], which consists of selected records of the complete KDD data set and has advantages over the original KDD data set. The dataset was divided into training and test dataset. Training is used to train the work presented here, while test dataset is used to test it. Test dataset contains additional attacks not described in training dataset. The attacks include the four most common categories of attack. Each attack type falls into one of the following four main categories [21].

- Denial of service (DoS) attacks; here, the attacker makes some computing or memory resource which makes the system too busy to handle legitimate requests. These attacks may be initiated by flooding a system with communications, abusing legitimate resources, targeting implementation bugs, or exploiting the system’s configuration.
- User to root (U2R) attacks; here, the attacker starts with accessing normal user account and exploits vulnerabilities to gain unauthorized access to the root. The most common U2R attacks cause buffer overflows.
- Remote to user (R2L) attacks; here, the attacker sends a packet to a machine, then exploits the machine’s vulnerabilities to gain local access as a user. This unauthorized access from a remote machine may include password guessing.
- Probing (PROBE); here, the attacker scans a network to gather information or find known vulnerabilities through actions such as port scanning.

In NSL-KDD dataset, these four attack classes are divided into 22 different attack classes that showed in Table 2.

**Table 2:** Attack type in Dataset

| *4 Main Attack Classes* | *22 Attack Classes* |
|---|---|
| DoS | back, land, neptune, pod, smurt, teardrop |
| R2L | ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster |
| U2R | buffer_overflow, perl, loadmodule, rootkit |
| Probing | ipsweep, nmap, portsweep, satan |

We utilize the NSL-Knowledge Discovery and Data Mining data set to test the system. We have taken 20% of the training dataset of NSL-KDD as input to the system. Due to the redundant records in the KDDCup'99 dataset the performance of the learning algorithms biased. The results of the accuracy and performance of learning algorithms on the KDDCup'99 data set are hence unreliable. NSL-KDD testing set provide more accurate information about the capability of the classifiers.

### 6.2 Performance Measures

The most widely used performance evaluations are Detection Rate (DR) and False Positive Rate (FPR). Good IDS must have high DR and a low or zero FPR. The performance of intrusion detection systems (IDS) are estimated by detection rates (DR). DR is defined as the number of intrusion instances detected by the system divided by the total number of intrusion instances present in the dataset.

$$DR = \frac{Total-detected-attacks}{Total-attacks} * 100 \quad (1)$$

$$FPR = \frac{Total-misclassified-attacks}{Total-normal-attacks} * 100 \quad (2)$$

### 6.3 Experiment and Analysis on Proposed Features

We perform two sets of experiments. From the first experiment, we examine the TPR and FPR of Naïve Bayes and Random Forest with all 41features. In this experiment for Dos Layer we selected features numbers 1,5,41, for R2L layer feature numbers 3,11,22,23,32,33 are selected, for U2R layer feature numbers 3,6,13,23,24,35,36 are selected, and probe layer feature numbers 3,6,12,30,32,35,36 are selected.

We choose 20% of the training dataset and 10-fold cross validation for the testing process. In 10-fold cross-validation, the available data is randomly divided into 10 disjoint subsets of approximately equal size. One of the subsets is then used as the test set and the remaining 9 sets are used for building the classifier. This is done repeatedly 10 times so that each subset is used as a test subset once. Table 3 shows performance of Naïve byes and Random Forest classifier with all features. From the results of this experiment we observe that Random Forest achieves higher attack TPR on all attack type. However the time taken to build model is very higher than Naïve Bayes.

**Table 3:** Performance of Naïve Bayes and Random Forest with 41 features

| Attack | Naïve Bayes | | Random Forest | |
|---|---|---|---|---|
| Classes | DR (%) | FPR (%) | DR (%) | FPR (%) |
| DoS | 95 | 0.02 | 100 | 0 |
| R2L | 46.4 | 0.014 | 100 | 0 |
| U2R | 100 | 0.06 | 100 | 0 |
| Probe | 87.7 | 0.054 | 100 | 0 |

For second experiment we examine the input data with selected attributes. The results are shown in Table 4. The performance of Naïve Bayes is obviously higher than first experiment.

**Table 4:** Performance of Naïve Bayes and Random Forest with selected features

| *Attack* | *Naïve Bayes* | | *Random Forest* | |
|---|---|---|---|---|
| *Classes* | *DR (%)* | *FPR (%)* | *DR (%)* | *FPR (%)* |
| DoS | 100 | 0.02 | 100 | 0.02 |
| R2L | 98 | 0.00 | 97 | 0.00 |
| U2R | 100 | 0.00 | 100 | 0 |
| Probe | 97 | 0.00 | 100 | 0.00 |

## 7. Conclusion and Future Work

In this paper we implement the system by analyzing the features of intrusion detection system. We presented an optimal intrusion detection system based on Naïve Bayes and Random Forest classifiers with feature selection. The system performed comparative analysis of two classifiers without features selection and with features selection. The main purpose of the system is to improve the performance of the system and classifiers using feature selection. The future works focus on applying the domain knowledge of security to improve the detection rates for current attacks in real time computer network, and ensemble with other mining algorithm for improving the detection rates in intrusion detection.

## References

[1] A. Boukerche and M.S.M.A. Notare, "Behavior-Based Intrusion Detection in Mobile Phone Systems," J. Parallel and Distributed Computing, vol. 62, no. 9, pp. 1476-1490, 2002

[2] A. Boukerche, K.R.L. Juc, J.B. Sobral, and M.S.M.A. Notare, "An Artificial Immune Based Intrusion Detection Model for Computer and Telecommunication Systems," Parallel Computing, vol. 30, nos. 5/6, pp. 629-646, 2004

[3] A. Boukerche, R.B. Machado, K.R.L. Juca´, J.B.M. Sobral, and M.S.M.A. Notare, "An Agent Based and Biological Inspired Real- Time Intrusion Detection and Security Model for Computer Network Operations," Computer Comm., vol. 30, no. 13, pp. 2649- 2660, Sept. 2007.

[4] A. Zainal, M.A. Maarol and S. M. Shamsuddin, "Ensemble Classifiers for Network Intrusion Detection System", *Journal of Information Assurance and Security*, 2009.

[5] Adebayo O. Adetunmbi, Samuel O. Falaki, Olumide S. Adewale and Boniface K. Alese, "Network Intrusion Detection Based On Rough Set And K-nearest

Neighbour", International Journal of Computing and ICT Research, Vol. 2 No. 1, June 2008.

[6] B. Lariviere, and D. Van den Poel, "Predicting Customer Retention and Profitability by Using Random Forests and Regression Forests Techniques." *Journal of Expert Systems with Applications*, Vol. 29, Issue 2, (August 2005) pp. 472-482.

[7] B. Yacine and C. Frederic, "Neural networks vs. decision trees for intrusion detection".

[8] Dr. R. L. Tulasi , M.Ravikanth "Impact of Feature Reduction on the Efficiency of Wireless Intrusion Detection Systems", *International Journal of Computer Trends and Technology- July to Aug Issue 2011.*

[9] http://www.cs.waikato.ac.nz/ml/weka

[10] Ibrahim H.E., Badr S.M, Shaheen M.A, "Adaptive Layered Approach using Machine Learning Techniques with Gain Ratio for Intrusion Detection Systems ", International Journal of Computer Applications (0975 – 8887) Volume 56– No.7, October 2012.

[11] J. Peters, B. De Baets, N.E.C Verhoest, R. Samson, S. Degroeve, P. De Becker and W. Huybrechts, "Random Forests as a Tool for Ecohydrological Distribution Modelling." *Journal of Ecological Modeling*, Vol 207, Issue 2-4, October 2007, pp. 304-318.

[12] J. Zhang, and M. Zulkernine, 2006. A Hybrid Network Intrusion Detection Technique Using Random Forests. In Proceedings of the IEEE First International Conference on Availability, Reliability and Security (ARES'06).

[13] Jain M., Richariya V. "An Improved Techniques Based on Naive Bayesian for Attack Detection", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 1, January 2012.

[14] L. Breiman, "Random Forests", *Machine Learning* 45(1):5–32, 2001.

[15] L. J. Hee, L. J. Hyouk, S.S. Gyoung , R.J. Ho and C.T. Myoung, "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System".

[16] L. Huaping, J. Yin, L. Sijia, "A New Intelligent Intrusion Detection Method Based on Attribute Reduction and Parameters Optimization of SVM", 2010 Second International Workshop on Education Technology and Computer Science.

[17] Naïve Bayes Classifier, Question and Answers.

[18] NSL-KDD dataset for network –based intrusion detection systems" available on http://iscx.info/NSL-KDD/

[19] P. Natesan, P. Balasubramanie, "Multi Stage Filter Using Enhanced Adaboost for Network Intrusion Detection", International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.3, May 2012.

[20] P. Xu, and F. Jelinek, "Random Forests and the Data Sparseness Problem in Language Modeling." *Journal of Computer Speech and Language*, Vol. 21, Issue 1(Jan 2007) pp. 105-152.

[21] S. A. Wafa and S. N. Reyadh, "Adaptive Framework for Network Intrusion by Using Genetic-Based Machine Learning Algorithm", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009.

[22] S. Lee, D. Kim, and J. Park, "A Hybrid Approach for Real-Time Network Intrusion Detection Systems", *International Conference on Computational Intelligence and Security*, 2007.

[23] Sharma el N., Mukherjee S.," A LAYERED APPROACH TO ENHANCE DETECTION OF NOVEL ATTACKS IN IDS", International Journal of Advances in Engineering & Technology, Sept 2012.

[24] V. P. Kshirsagar, Dharmaraj R. Patil, "Application of Variant of AdaBoost-Based Machine Learning Algorithm in Network Intrusion Detection", International Journal of Computer Science and Security (IJCSS), Volume (4): Issue :( 2).

## Author Profile

**Aye Mon Yi** received the B.C.Sc. (Hons:) degree from Computer University (Myeik) in 2004 and M.C.Sc. degree from University of Computer Studies, Yangon (UCSY) in 2006. Now she is a PhD candidate at University of Computer Studies, Mandalay. Her interested fields are Data Mining, Machine Learning and Network Security.