IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 4, Ver. VII (Jul – Aug. 2014), PP 16-26
www.iosrjournals.org

An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment

Partha Ghosh[1], Chameli Debnath[2], Dipjyoti Metia[3], Dr. Ruma Dutta[4]
[1,2,4]*(Information Technology, Netaji Subhash Engineering College/West Bengal University of Technology, India)*
[3]*(Control & Instrumentation,Netaji Subhash Engineering College/West Bengal University of Technology, India)*

Abstract: *Cloud Computing offers latest computing paradigm where application, data and IT services are provided online over the Internet. One of the significant concerns in Cloud Computing is security. Since data is exposed to many users, security and privacy have become the key issues of Cloud Computing. Intrusion Detection System (IDS) plays an important role to identify intrusions by monitoring the activity of the system and alert the user about malicious behaviours and detect attacks. To detect those attacks, several classification methods have been used till now. This paper deals with Intrusion Detection System by the method of classification. In this paper, KNN is applied as binary classifier for anomaly detection. Neural Network is applied for detecting abnormal classes after KNN classification. Before classification, feature selection has been used to select relevant features. For our experimental analysis, we have used NSL-KDD dataset where all samples of "KDDTrain+" used as training dataset and "KDDTest+" samples are used as testing dataset. We use Rough Set Theory and Information Gain to select relevant features. Experimental results show that, we get better accuracy with our proposed hybrid KNN_NN classifier model for Intrusion Detection.*

Keywords: *Information Gain (IG), Intrusion Detection System (IDS), K-Nearest Neighbor (KNN), Neural Network (NN), NSL-KDD dataset, Rough Set Theory (RST).*

I. Introduction

Cloud Computing refers to applications and services, in which data centers on the world, is made available to users who have an Internet connection. Cloud Computing enables users to store and process data over the Internet. It can deliver both software and hardware as on demand resources and services [1]. With our growing technologies, information is the most precious asset for all and the risk of unauthorized access of data is increased continuously. Intrusion detection and prevention in Cloud infrastructure is the major concern after data security. It is necessary to provide security for information. Network security is very important as networks are exposed to an increasing number of security threats. Intrusion Detection System (IDS) plays an important role to provide security and reduce damage of the information system and for the network and computer security system [2]. It monitors various events in a system or network, to determine whether an intrusion has occurred or not and alerts the system or network administrator against malicious behaviors or detected attacks. Intrusion Detection System (IDS) can be classified into two types according to environment of the IDS: Host Based Intrusion Detection System (HIDS) and Network Based Intrusion Detection System (NIDS). HIDS monitors and analyzes the information collected from a specific host machine [3]. It reports the existence of attack depending upon detection of deviation from expected behavior. Each HIDS detects intrusion for the machines in which it is placed. HIDS cannot monitor the application themselves; it can only monitor the resource usage of the application [4]. NIDS has stronger detection mechanism to detect network intruders. It can detect network intrusion by comparing current behavior with already observed behavior. NIDS has very limited visibility inside the host machines [3].There are mainly two approaches of IDS, namely misuse detection and anomaly detection [5]. In misuse detection, it only detects those attacks for which a signature has previously been created, i.e., it uses pattern matching to detect known attack patterns. Misuse detection also refers to signature based detection, pattern matching, rule-based detection. Anomaly detection can be used to detect unknown attacks at different levels [3]. It can detect any action that significantly deviates from the normal behavior i.e., it looks for anomalies. It is able to detect unknown attacks based on audit. Anomaly detection uses predefined concepts about 'normal' and 'abnormal' system activity to differentiate anomalies and normal system behavior and to block anomalies [6]. In our experiment, we have used NSL-KDD dataset for evaluation of the Intrusion Detection System which is an improved version of original KDD dataset. NSL-KDD dataset solved some inherent problems of original KDD dataset. Our aim in this work is to filter out all redundant features and worthless information from the dataset, to select only the relevant features and detect intrusion. For feature selection we have used two methods. Those are Rough Set Theory as a Wrapper-based method and Information Gain as a Filter-based method. After that, we have applied our proposed hybrid classification algorithm to enhance the detection accuracy of the IDS. Rough Set is a wrapper-based feature selection method used to approximately or "roughly" define equivalent classes. It reduces the computational complexity of learning process and eliminates redundant or irrelevant attributes. In case of Information Gain, which is used as a filter-based feature selection method, is measures the amount of information in bits about the class prediction. For classification, we have applied K-Nearest Neighbor (KNN) as a binary classifier for anomaly detection and Neural Network (NN) for misuse detection. In KNN classification, an object is classified by a majority vote of its neighbors. The object is consequently assigned to the class that is most common among its Kth Nearest Neighbor, where K is a positive integer that is typically small. After using KNN classifier as a binary class classification method, we again used another classification method for abnormal classes of KNN to find out specific attack type classes. In our experiment, we have used Back Propagation Neural Network for classification of abnormal classes by transforming a set of inputs to a set of desired outputs.

II. Related Works

In 2009, Neveen I. Ghali [7] has presented a new hybrid algorithm RSNNA (Rough Set Neural Network Algorithm), used to significantly reduce a number of computer resources, both memory and CPU time, and required to detect attacks. The algorithm uses Rough Set Theory in order to select out feature reducts and a trained Artificial Neural Network to identify any kind of new attacks. In this algorithm, using Rough Set Theory from 6385 samples(6128 sets for training and 257 sets for testing) of KDD99 dataset only 7 features are selected among 41 features. The proposed model gives better and robust representation of data as it was able to select features, reduce time and also it reduces error in detecting new attacks. In 2010, Rupali Datti et al. [8] proposed Linear Discriminant Analysis. Linear Discriminant Analysis (LDA) algorithm is used to extract features for detecting intrusions and Back Propagation algorithm is used for classification of attacks. In this experiment authors have used 11850 training samples, 9652 sets of test samples with 41features of NSL-KDD dataset. It was able to reduce the 41 features of the dataset into 4 features and as a result it gives reduction of the input data, hence training time also reduced. In 2010, Shilpa Lakhina et al. [9] used a new hybrid algorithm PCANNA (Principal Component Analysis Neural Network Algorithm) where Principal Component Analysis (PCA) is used as a reduction tool. The experiment with NSL-KDD dataset gives better and robust representation of data and it was able to reduce features. This experiment using PCANNA, training time and testing time are reduced. In 2010, N. Suguna et al. [10] proposed a method where Genetic Algorithm (GA) and K-Nearest Neighbor (KNN) are combined to improve classification performance. In this paper, instead of considering all the training samples and taking K-neighbors, the GA is employed to take K-neighbors straightway and then the distance to classify the test samples is calculated. Before classification, features are selected using Rough Set Theory, hybrid with Bee Colony Optimization (BCO) techniques. The experimental result shows that the proposed method not only reduces the complexity of the KNN, it also improves the classification accuracy. In 2011, Al-Janabi et al. [11] developed an anomaly-based Intrusion Detection System which can quickly detect and classify various attacks. They worked with packet behaviour as parameters in anomaly detection. They have used Back Propagation Artificial Neural Network to learn system's behaviour. KDD'99 data set is used in their experiment and the obtained results satisfy their work objective. In 2013, Rowayda A. Sadek et al. [12] proposed a new hybrid algorithm NNIV-RS (Neural Network with Indicator Variable using Rough Set for attribute reduction) algorithm, which reduces the amount of computer resources like memory and CPU time required to detect attack. In this algorithm, Rough Set is used to select out feature reducts, indicator variable is used to represent dataset in more efficient way and Neural Network is used as a network traffic packet classification. This algorithm gives better and robust representation of data. It was able to select significant attributes from the selected features and achieve detection accuracy with a false alarm rate of 3%.In 2013, Deeman Y. Mahmood et al. [13] applied K-star algorithm with filtering analysis to build a Network Intrusion Detection System. They have used NSL-KDD dataset with a split of 66.0% for the training set and the remaining for the testing set and WEKA toolkit was used in the testing process. As a result, it gives very accurate result with low false positive rate and high true positive rate and it takes less learning time in comparison with other existing approaches which are used for efficient network intrusion detection. In 2013, Yogita B. Bhavsar et al. [14] used Support Vector Machine for classification. Verification of the effectiveness of the proposed system is done by conducting some experiments using NSL-KDD CUP'99 dataset. The drawback of using SVM classification is its extensive training time. In this proposed system, by performing proper data pre-processing and by using proper SVM kernel selection i.e., radial basis function (RBF), extensive time required to build SVM model is reduced. When they have performed experiment with 10 fold cross validation and Gaussian RBF kernel of SVM, it shows that the attack detection accuracy increases when they have changed classification to 10 fold cross validation and re-evaluation using supplied test set with same RBF SVM kernel function.

III. Nsl-Kdd Dataset

This paper has used NSL-KDD dataset for the experiment. NSL-KDD dataset are based on the original KDD dataset. NSL-KDD dataset consists of four components, ”KDDTrain+”,”KDDTest+”, ”20%KDDTraining+” and “KDDTest-21” [15]. For our experimental purpose, we have used “KDDTrain+”

and “KDDTest+” which consists of 125,973 and 22,544 records respectively. Each component contains 41 features labeled as normal or specific attack type. These 41 features grouped into 4 categories: basic features, content features, time-based traffic features and host-based traffic features. The value of these features mainly based on continuous, discrete and symbolic value. All the feature categories are discussed below:

- **Basic features:** Among 41 features of NSL-KDD dataset, the features which derived from TCP/IP connection are known as basic features. These features lead an implicit delay in detection (example: duration, protocol_type, service etc).
- **Content features:** Some features use domain knowledge to access the payload of the original TCP packets are known as content features (example: logged_in, root_shell, is_hot_login etc).
- **Time-based traffic features:** Time-based traffic features are specially designed to capture one special property of the dataset that is to capture those features which are mature over a 2 second temporal window (example: srv_serror_rate, srv_rerror_rate).
- **Host-based traffic features:** Among four types of attacks in NSL-KDD dataset, few attacks span longer than 2 seconds intervals. Host-based traffic features are designed to access all attacks which span longer than 2 second intervals that have the same destination host as the current connection (example: serror_rate, rerror_rate).

NSL-KDD dataset mainly contain four types of attacks. They are given below:

- **Denial of Service attack (DoS):** When an attacker successfully makes computing and memory resources too busy or denies legitimate users access, to a machine is called DoS attack.
- **Remote to Local attack (R2L):** In R2L attack, an attacker wants to gain the local access as a user of particular machine without any account. To accomplish this, attacker sends packet to a remote machine over a network and exploits some vulnerability.
- **User to Root attack (U2R):** By using these types of attack, attacker can access normal user account on the system and gain the root access to the system and exploit some vulnerability.
- **Probe:** In this type of attack, by scanning a network of computer attacker can gather all necessary information about the target system or can find out known vulnerabilities.

Statistical records in NSL-KDD dataset:

NSL-KDD dataset	DoS	Probe	R2L	U2R	Normal	Total records
KDDTrain+	45927	11656	995	52	67343	**125973**
20%KDDTraining+	9234	2289	209	11	13449	**25192**
KDDTest+	7458	2421	2554	400	9711	**22544**
KDDTest-21	4342	2402	2554	400	2152	**11850**

Reason behind using NSL-KDD dataset: NSL-KDD dataset is the new version of original KDD dataset. KDD dataset [16] is the most widely used dataset for the intrusion detection. But from the statistical analysis of the KDD dataset, researchers found that it affects the performance highly and it gives very poor evaluation of the system. To solve the problem of KDD dataset, NSL-KDD dataset was proposed. It solved some inherent problem of the KDD dataset. NSL-KDD dataset consists of selected records of KDD dataset. Advantages of using NSL-KDD dataset over original KDD dataset are given below:

- It eliminates redundant records from the training set and also removed duplicate records from the testing set to improve the intrusion detection performance.
- In NSL-KDD dataset, the number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD dataset.
- NSL-KDD dataset consists of reasonable numbers of instances both in the training and testing set. For this reason it is suitable for experiment purposes with the complete set and there is no need to randomly select a small portion.
- In case of classification NSL-KDD dataset gives an accurate evaluation for different learning techniques.

IV. Proposed Model

Cloud Computing is an internet based computing, where infrastructure, information and application are provided based on demand. Security and privacy are the major concerns in Cloud Computing. To overcome this, Intrusion Detection System (IDS) is used to detect whether an intrusion has occurred or not. By monitoring the activity of the system, IDS detect attacks and alert users about detected attacks. Anomaly detection and misuse detection are main approaches of IDS. Anomaly detection can be used to detect unknown attacks and it can detect any action that significantly deviates from the normal system behavior. Misuse detection can detect only those attacks for which a signature has previously been created and applying pattern matching to detect known attack patterns. In this section, we first present the whole framework of our experiment architecture and then we discuss main models. In our experiment to detect intrusion we have used NSL-KDD dataset which consists of large number of records. To reduce irrelevant records from the dataset and to select significant features for better experimental result Rough Set Theory and Information Gain are used as feature selection method. The proposed intrusion detection techniques initially select features from the training dataset using wrapper-based method (Rough Set Theory) and filter-based method (Information Gain). After selecting the features, by using both Rough Set Theory and Information Gain, the dataset is passed to our proposed hybrid model for classification. In our hybrid KNN_NN two layer model we use both the concept of K-Nearest Neighbor and Neural Network classifier. KNN is used for binary classification, which classify data into 'normal' and 'abnormal' classes. Then the 'abnormal' classes of KNN are passed to Neural Network for classifying specific attack type. In our proposed system, we first properly train the dataset and after that test dataset is applied for classification. Figure 1 shows our proposed model of Intrusion Detection System.

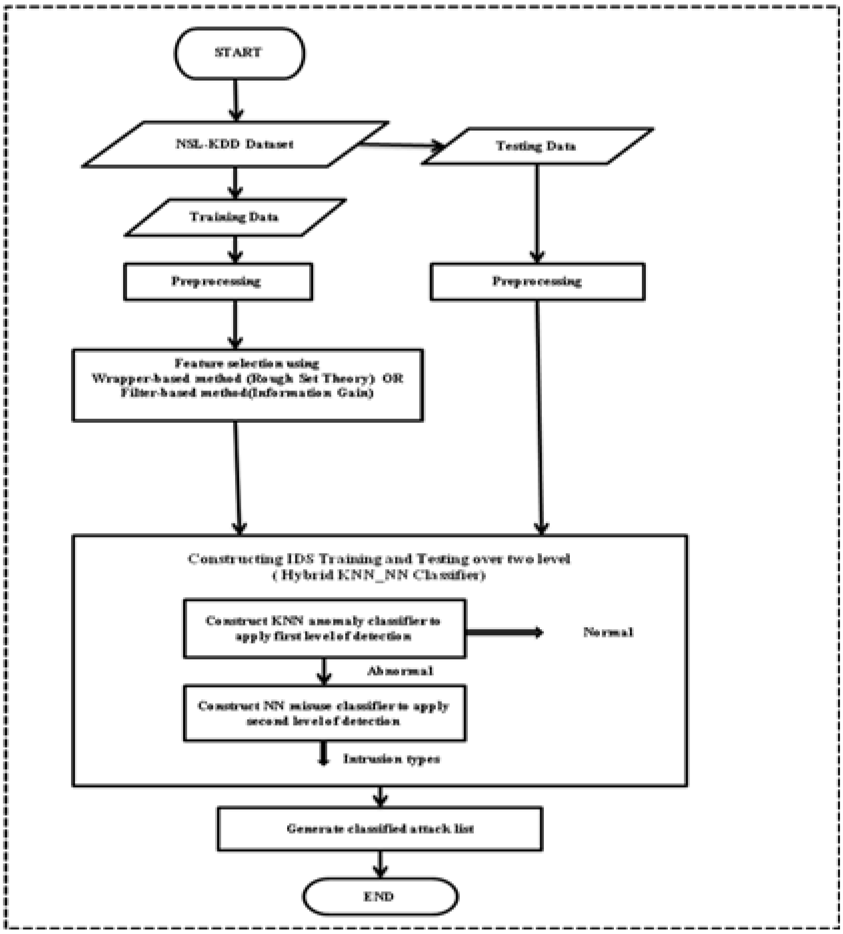
START
NSL-KDD Dataset
Testing Data
Training Data
Preprocessing
Preprocessing
Feature selection using Wrapper-based method (Rough Set Theory) OR Filter-based method(Information Gain)
Constructing IDS Training and Testing over two level (Hybrid KNN_NN Classifier)
Construct KNN anomaly classifier to apply first level of detection
Normal
Abnormal
Construct NN misuse classifier to apply second level of detection
Intrusion types
Generate classified attack list
END

Figure1: Flow of Intrusion Detection System

All the different steps of the Intrusion Detection System (IDS) are discussed below in detail:

- **Dataset preprocessing:** Data pre-processing is necessary to make NSL-KDD dataset as suitable input for our experiment. In NSL-KDD dataset each connection of the dataset contains 42 features including one decision feature for normal or specific attack type. Among 41 features three conditional features (protocol_type, service, flag) and one decision making feature have non-numeric value. So we need to convert these non-numeric values into numeric values. In case of three conditional features we have assigned numeric values by replacing non-numeric values based on the occurrence frequency of the particular feature type [17]. As an example, in 'protocol_type' feature, before data pre-processing there are three types of value: 'tcp', 'udp' and 'icmp'. Among these three, tcp have the greater number of occurrence. So we replaced 'tcp' by 3, 'udp' by 2, because 'udp' has second repeatable priority and so on. For decision making feature we have assigned 1 for 'DoS' attack, 2 for 'normal', 3 for 'probe' and so on, based on their alphabetic order.
- **Feature selection:** Feature selection is a process of finding a subset of significant features from the original set of features. Feature selection reduces the number of irrelevant redundant features from the dataset and

selects significant features to improve the detection accuracy. Redundant features are those features which closely correlated with one or more features and provide no more information than the selected relevant features. For constructing high performance Intrusion Detection System, effective feature selection is very important. Removing irrelevant redundant features from the input dataset improves the performance of the classification and also decreases storing of memory space. In fact, the performance and accuracy may improve due to removing irrelevant features and also training time of dataset also be reduced [18].There are mainly two methods of feature selection: Wrapper-based method and Filter based method. We used Rough Set Theory (RST) as a Wrapper-based method using RSES toolkit [19] and Information Gain (IG) as a Filter-based method using WEKA (Waikato Environment for Knowledge Analysis) toolkit [20].

(i) Wrapper-based method: In Wrapper-based method, subset selection takes place based on the learning algorithm. That means wrapper-based method requires one predetermined learning algorithm in feature selection. It uses to evaluate performance and determine which features are selected. It gives superior performance as it finds features better suited to the predetermined learning algorithm, but computationally it is more expensive. This method based on successive elimination of features and leaving those features that lead to highest accuracy. Main objective of wrapper-based method is to search for the best subset of features. To achieve the best possible performance with a particular learning algorithm on a particular training set, a feature subset selection method should consider how the algorithm and the training set interact [21]. Before applying Rough Set as a wrapper-based method for feature selection, we have to perform normalization and discretization on the data set.

Normalization: All the conditional attributes of the NSL-KDD dataset may contain discrete as well as continuous value. Combination of discrete and continuous values of these features makes the range of the features values is different. To solve this problem we normalized the dataset and as a result range of all features are same and also all features are comparable. For normalization of the dataset we have used min-max normalization [22]. Here we assume min_A and max_A are the minimum and maximum values of the feature A, where a value v of A maps to v' within the range of new_min_A to new_max_A. In this normalization process we have assigned the value of new_min_A is 0.0 and new_max_A is 1.0 respectively. Following formula is used for the min-max normalization:

$$v' = \frac{\mathrm{v} - \mathrm{min}_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Discretization: Discretization techniques can be categorized based on how the discretization is performed, such as whether it uses class information or which direction it proceeds (i.e., top-down vs. bottom-up). If the discretization process uses class information, then we say it is supervised discretization. Otherwise, it is unsupervised. If the process starts by first finding one or a few points (called split points or cut points) to split the entire attribute range, and then repeats this recursively on the resulting intervals, it is called top-down discretization or splitting [23]. This contrasts with bottom-up discretization or merging, which starts by considering all of the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, and then recursively applies this process to the resulting intervals. Discretization simplifies the original dataset. Before applying Rough Set Theory, we have performed normalization and discretization process on our dataset.

Rough Set Theory (RST): Rough Set Theory reduces the computational complexity of learning process and eliminates the unimportant or irrelevant attributes so that the knowledge discovery in datasets can be efficiently learned [24]. There are several subsets of attributes in Rough Set, among them minimal subsets are called reducts [25]. Rough Set Theory deals with inconsistencies, uncertainty and incompleteness by imposing an upper and lower approximation to set membership. The Rough Set approach to processing of incomplete data is based on the lower and the upper approximation. Every subset defined through upper and lower approximation is known as Rough Set.

- Upper Approximation : $\overline{R}X = \cup \{Y \in U/R : Y \cap X \neq \emptyset\}$
- Lower Approximation : $\underline{R}X = \cup \{Y \in U/R : Y \subseteq X\}$

Four basic classes of Rough Set Theory:

- X is roughly B-definable, iff $\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) \neq U$
- X is internally B-undefinable, iff $\underline{B}(X) = \emptyset$ and $\overline{B}(X) \neq U$
- X is externally B-undefinable, iff $\underline{B}(X) \neq \emptyset$ and $\overline{B}(X) = U$
- X is totally B-undefinable, iff $\underline{B}(X) = \emptyset$ and $\overline{B}(X) = U$

Definitions which show the reduct derivations [26] are as follows:

Definition1: In Rough Set Theory knowledge is represented by means of a table called an information system given by $S =< U, A, V, f >$; where $U = \{x_1, x_2, \dots, x_n\}$ is a finite set of objects of the universe (n is the number of objects); A is the non empty finite set of features, $A = \{a_1, a_2, \dots, a_m\}$; $V = \cup_{a \in A} V_a$ and V_a is a domain of feature a; $f : U \times A \rightarrow A$ is a total function such that $f(x, a) \in V_a$ for each $a \in A, x \in U$. If the features in A can be divided into condition set C and decision feature set D; i.e. $A = C \cup D$ and $C \cap D = \emptyset$. The information system A is known as decision system or decision table.

Definition2: In case of Rough Set, every $B \subseteq A$ yields an equivalence relation up to indiscernibility, $IND_A(B) \subseteq (U \times U)$, given by: $IND_A(B) = \{(x, x'):\forall a \in B\ a(x) = (x')\}$ a reduct of A is the least $B \subseteq A$ that is equivalent to A up to indiscernibility i.e., $IND_A(B) = IND_A(A)$.

(ii) Filter-based method: In Filter-based method, the subset selection procedure is independent of the learning algorithm [27]. That means filter-based method relies on general characteristics of the training data to select some features without involving any learning algorithm, therefore it does not inherit any predetermined learning algorithm [28]. Filter-based method select features according to some evaluation criterion (for example, correlation between the feature and the class knowledge). In filter-based method we rank each feature and select the highest ranking features. This method is very efficient and fast for computing. When the number of features becomes very large, the filter-based method is used to its computational efficiency. In our paper we rank our features depending on their Information Gain.

Information Gain (IG): Information Gain (IG) measures the amount of information in bits about the class prediction. For each dataset we select the subset of features based on their rank value. It also measures the expected reduction in entropy. Entropy is commonly used in the information theory measure, which characterizes the purity of an arbitrary collection of examples. Entropy is in the foundation of the Information Gain attribute ranking methods. The entropy measure is considered as a measure of system's unpredictability. In Information Gain the features are filtered to create the most significant feature subset before the learning process. The computation of the information gain of a set of attributes with respect to all classes calculated [29] [30] as follows: Let, the set of training set sample of the dataset is S with their representing labels. We consider there are n classes and the training set consists of S_i samples of class I and S is the total number of sample S in the training set. So required information needed to classify a given sample is calculated by [31],

$$I(S_1, S_2 \dots S_n) = \sum_{i=0}^{n} \frac{S_i}{S} \log_2 \frac{S_i}{S}$$

Feature F with values $\{f_1, f_2, \dots f_v\}$ can divide the training dataset into $\{S_1, S_2, S_3, \dots S_v\}$ subsets where S_j is the subset which has the value f_j for the feature F. Moreover let S_j contain S_{ij} samples of class i. Entropy of the feature F is given below:

$$E(F) = -\sum_{i=0}^{n} \frac{S_{1j}+S_{2j}+S_{3j}+\cdots S_{ij}}{S} I(S_{1j}, S_{2j}, S_{3j}, \dots S_{ij})$$

Information Gain for F can be calculated using following formula:

$$Gain(F) = I(S_1, S_2 \dots S_n) - E(F)$$

For selecting features from NSL-KDD dataset, first we used Rough Set Theory as a wrapper-based method and select 25 features using RSES toolkit. Then Information Gain is used as a filter-based feature selection method using WEKA toolkit and here we again select 25 high rank features as 25 features are selected by Rough Set Theory. After selecting features all the datasets are applied separately for classification.

- **Classification:** Classification may refer to categorization. It is a process in which ideas and objects are recognized, differentiated and understood. In classification, records of dataset are divided into training and test datasets. In other case, we can use train component of dataset as training dataset and test component of dataset as testing dataset. In our experiment, we have used "KDDTrain+" as training dataset and "KDDTest+"as testing dataset for classification. There are several classification algorithms; among them we have used the concept of KNN and Neural Network classifier for our experiment.

(i) KNN (K-Nearest Neighbor): In classification the set of possible classes are defined with respect to a specific classification task. KNN or K-Nearest Neighbor classification classifies instances based on their similarity to instances in the training data. KNN is one of the top 10 data mining algorithm and it makes decision based on the entire training data set. K-Nearest Neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (example: distance functions). KNN has been used in statistical estimation and pattern recognition. A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K-Nearest Neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor. It is also a non parametric lazy algorithm. It means that it does not use the training data points to do any generalization. In other words, there is no explicit training phase and it is very minimal. This means the training phase is reasonably fast. Lack of generalization means that KNN keeps all the training data or more exactly we can say, all the training data is needed during the testing phase. Whereas parametric means that it does not make any assumptions on the underlying data distribution. The K-Nearest Neighbor (KNN) algorithm is among the simplest of all machine learning algorithms [32] [33]. The KNN algorithm is first implemented by introducing some notations $S = (x_i, y_i)$; i=1,2,…N is considered the training set, where x_i is the d-dimensional feature vector, and $y_i \in \{+1, -1\}$ is associated with the observed class labels. For simplicity, we consider a binary classification. We generally suppose that all training data are samples of random variables (X, Y) with unknown distribution. With previously labeled samples as the training set S , the KNN algorithm constructs a local sub region $R(x) \subseteq \Re^d$ of the input space, which is situated at the estimation point x . The predicting region $R(x)$ contains the closest training points to x , which is written as follows: $R(x) = \{\hat{x} | D\ (x, \hat{x}) \leq d_{(k)}\}$, where $d_{(k)}$ is the k^{th} order statistic of $\{D(x, \hat{x})\}_1^N$, and $D(x, \hat{x})$ is the distance metric. $k[y]$ denotes the number of samples in region $R(x)$, which is labeled by y . The KNN algorithm is statistically designed for the estimation of posterior probability $p(y \mid x)$ of the observation point x :

$$p(y \mid x) = \frac{P(x \mid y)\, P(y)}{P(x)} \cong \frac{k[y]}{k}$$

For a given observation x, the decision $g(x)$ is formulated by evaluating the values of $k[y]$ and selecting the class that has the highest $k[y]$ value:

$$g(x) = \begin{cases} 1, k[y = 1] \geq k[y = -1], \\ -1, k[y = -1] \geq k[y = 1] \end{cases}$$

Thus, the decision that maximizes the associated posterior probability is employed in the KNN algorithm. For a binary classification problem in which $y_i \in \{+1, -1\}$, the KNN algorithm produces the following decision rule:

$g(x) = sgn\ (ave_{x_i \in R(x)}\, y_i)$.

(ii) Neural Network (NN): The aim of using Neural Network for intrusion detection is to classify data as normal or a specific attack. It consists of highly interconnected processing elements. Neural Network transforms a set of inputs to a set of desired outputs. Neural Network is able to adapt to the desired outputs by modifying the connections between the nodes [34]. The ability of high tolerance makes the Neural Network flexible and powerful in IDS [35].

The advantages of using Neural Networks are: [36]

- Neural Networks are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model.
- Neural Network has the capability of analysing the incomplete and distorted data from the network.
- Neural Networks are universal functional approximators. The Neural Networks can approximate any function with arbitrary accuracy. Since any classification procedure attempts a functional relationship between the group membership and the attributes of the object, accurate identification of this underlying function is important.
- Neural Networks are non linear models, which makes them flexible in modelling real world complex relationships.
- Neural Networks are able to estimate the posterior probabilities, which provide the basis for establishing classification rule and performing statistical analysis.

In our experiment, after applying KNN as binary classification for normal and abnormal classes, we again apply Back Propagation Neural Network classifier for classifying abnormal classes. Back propagation is a systematic method of training multi-layer Artificial Neural Networks. It consists of at least three layers of units named; an input layer, intermediate hidden layer and an output layer. These units are connected in a feed-forward fashion with input units fully connected to units of hidden layer and hidden units fully connected to unit of output layer. We have used 41 features of dataset as input units and one hidden layer which consist of 21 neurons and give 1 output in output layer. The output of the Back Propagation Neural Network is interpreted as a classification decision. In our experiment we have used MATLAB R2014a (version 8.2) software which was released on March 2014. In Back Propagation Neural Network we feed training patterns one by one to the neunet (a network of neurodes), and then we update the weights depending on the error. The back propagation procedure is said to perform a 'gradient descent' on the error surface in the weight space where the weights change in the direction of the greatest rate of decrease of error [37]. Figure 2 shows the learning mechanism of Neural Network.

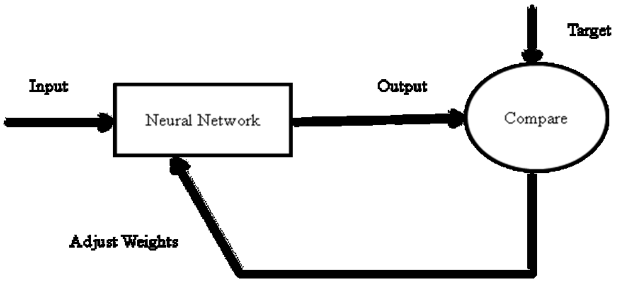

Figure 2: Learning mechanism of Neural Network.

In a multi-layer neunet, for a given training pattern, error $e(G,K)$ in the output of neurode $u(G,K)$, for $1 \leq K \leq N(G)$, is defined to be:$e(G,K) = A'[I(G,K)] \times [z(G,K) - y(G,K)]$ in which $[z(G,K) - y(G,K)]$ is the difference between the desired output $z(G,K)$ and the observed output $y(G,K)$. The derivative $A'[I(G,K)]$ gives the rate of change of the activation function A at $I(G,K)$, the value input to neurode $u(G,K)$. We have used Quasi Newton BFGS learning method in our Back Propagation Neural Network. For learning method, Newton's method is an alternative to the conjugate gradient methods for fast optimization [38]. The basic step of Newton's method is: $x_{k+1} = x_k - A_k^{-1} g_k$, where A_k is the Hessian matrix (second derivatives) of the performance index at the current values of the weights and biases. Newton's method often converges faster than conjugate gradient methods. Unfortunately, it is complex and expensive to compute the Hessian matrix for Feed Forward Neural Networks. There are number of algorithms that is based on Newton's method, but which doesn't require calculation of second derivatives. These are called Quasi-Newton (or Secant) methods. They update an approximate Hessian matrix for all iteration of the algorithm. The update is computed as a function of the gradient. This algorithm has been implemented in the trainbfg routine. Quasi Newton BFGS learning method requires more computation for all iteration and more storage than the conjugate gradient methods, although it generally converges in less iteration.

In our experiment we used feature selection to minimize memory space required for large dataset and to reduce learning time. Also, to produce much better result, we have designed hybrid KNN_NN model instead of using separated KNN and Neural Network classification method.

- **Algorithm of our proposed Hybrid KNN_NN model for classification:** In our proposed hybrid KNN_NN model, we have used KNN classification method for anomaly detection as a binary classifier. After that Neural Network classifier is used for misuse detection to detect specific attack type of abnormal classes.

Algorithm (1): Suggested-KNN
Input: NSL-KDD for training and testing.
Output: Results of anomaly detection on NSL-KDD testing dataset using KNN.
Steps:

1. Suppose there are j training categories $C_1, C_2, \dots, C_j$ and the sum of the training samples is N, after feature reduction, they become m-dimension feature vector.
2. Make sample X to be the same feature vector of the form $(X_1, X_2, \dots, X_m)$, as all training samples.
3. Calculate the similarities between all training samples and X. Taking the i^{th} sample $d_i(d_{i1}, d_{i2}, \dots, d_{im})$ as an example; the similarity $SIM(X, d_i)$ is calculated using Euclidean distance.
4. Choose k samples which are larger from N similarities of $SIM(X, d_i)$, $(i = 1, 2, \dots, N)$, and treat them as a KNN collection of X. Then calculate the probability of X belong to each category respectively.
5. Judge sample X to be the category which has the largest probability $P(X, C_j)$.
6. End.

Algorithm (2): Suggested-NN
Input: Abnormal classes produced by KNN from NSL-KDD dataset.
Output: Results of misuse detection on NSL-KDD testing dataset using NN (Neural Network).
Steps:
Main assumption for the training process:
Learning method : Quasi Newton BFGS.
Number of Epochs: 1000.
MSE (Mean Square error): 0.01.
Learning rate : 0.9.

Activation function: tansiq and purelin.
Number of neurons in the Input layer: (41 for NSL-KDD dataset, 25 after feature selection using Rough Set and 25 after feature selection using Information Gain).
Number of neurons in the hidden layers: (21 for 41 input neurons, in case of features selected by Rough Set Theory and Information Gain 13 for 25 input neurons i.e., half of input neurons).
Number of neurons in the output layer: (No. of intrusion classes are 4).
Train and Test on NSL-KDD, to construct final NN misuse detection result.
End.

V. Results & Analysis

In order to evaluate the performance of proposed algorithm, we used NSL-KDD benchmark dataset. All experiments were performed using an Intel Core i7 processor with 8 GB of RAM and windows 7 operating system. Using this high configured machine, at first we have applied Rough Set Theory and Information Gain on NSL-KDD dataset for feature selection. To perform Rough Set Theory we used RSES toolkit and for Information Gain WEKA toolkit is used. For classification purpose, we have used MATLAB R2014a (version 8.2) software. In our experiment, it shows that classification with 41 features of NSL-KDD dataset and classification after feature selection, using Rough Set Theory and Information Gain, produced approximately same or better result. To produce better experimental result and to reduce memory space and learning time, it is better to use dataset after feature selection. For feature selection we used Rough Set Theory as wrapper based method and select 25 features among 41 features of NSL-KDD dataset. Information Gain is also used as a filter based feature selection method and we have select high ranking 25 features as 25 features are selected by using Rough Set Theory, to identify which feature selection method is better among this two. In our experiment, feature selection using Information Gain gives better result than Rough Set Theory. In classification, we have used both KNN and Neural Network classification separately on 41 features of NSL-KDD dataset and on 25 features selected by Rough Set Theory and Information Gain. Figure 3 show that, KNN classification used for 41 features of NSL-KDD dataset produced same result with 25 features of Information Gain. But both of them give better result than applying 25 features of Rough Set Theory. In case of Neural Network classification, 25 features of Rough Set Theory produced much better result than Information Gain and 41 features of the NSL-KDD dataset.

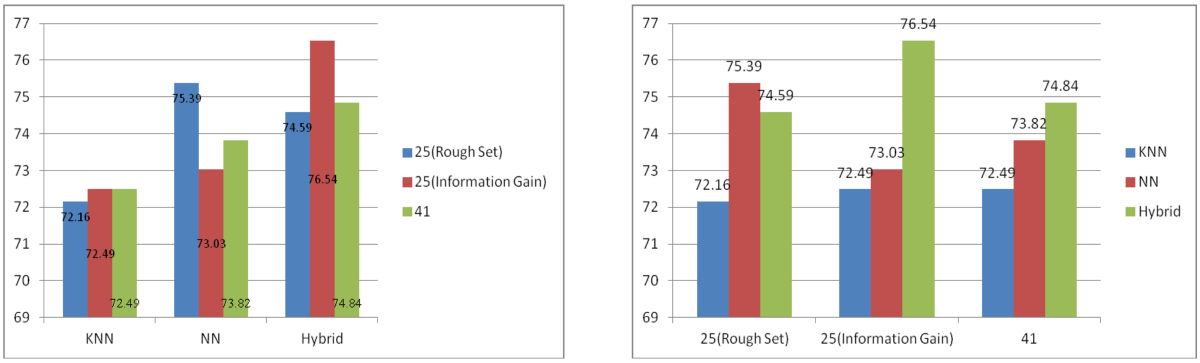

Figure 3: Detection results of test dataset with 41 and 25 features.

To produce better result than using KNN and Neural Network classification separately, we have designed our proposed hybrid KNN_NN multilevel classification model. In this hybrid model, KNN classification is used for anomaly detection as a binary classifier with 'normal' and 'abnormal' classes. After that Neural Network classifier is used as misuse detection to detect specific attack type of 'abnormal' classes. In our proposed hybrid KNN_NN classification model, Information Gain on 25 features of the NSL-KDD dataset produced better result as compared to using 41 features of the NSL-KDD dataset and 25 features of Rough Set Theory. Figure 4 shows the result of KNN, NN and hybrid KNN_NN classification method after applying Rough Set Theory and Information Gain on NSL-KDD dataset for feature selection. In most cases, our proposed KNN_NN hybrid model generates better result than separately used K-Nearest Neighbor and Neural Network classifier.

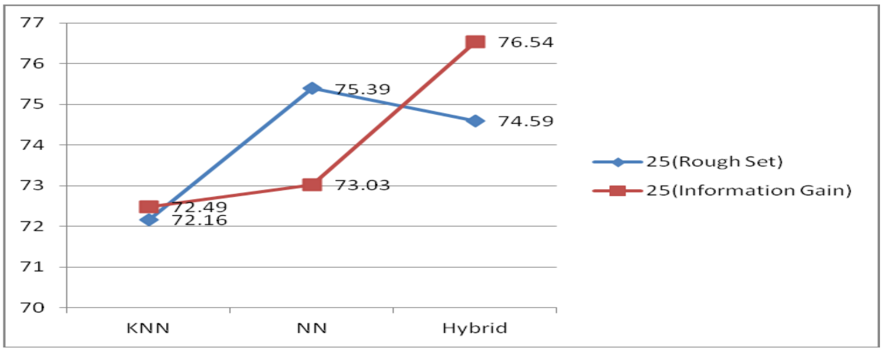

Figure 4: Results of detection and classification with Rough Set and Information Gain.

In our proposed KNN_NN hybrid model, Information Gain applied on NSL-KDD dataset as a filter-based feature selection method, gives much better result then others because the filter-based method is very efficient and fast computing technique for feature selection when the number of features becomes very large in the dataset. The abnormal classes of NSL-KDD dataset consists of four classes (Dos, Probe, R2L and U2R). The training ('KDDTrain+') and testing component ('KDDTest+') of NSL-KDD dataset used for our experiment produced good cross validation result for all classes. But in classification using testset, it produced good classification result only for DoS and Probe classes. In case of R2L and U2R classes, they are misclassified as testing dataset of NSL-KDD dataset contains greater number of R2L and U2R classes as compared to training dataset. To get better classification result for R2L and U2R classes, NSL-KDD test dataset should be improved. Our proposed hybrid multilevel KNN_NN classification model is more efficient than separately used KNN and NN classification method. In our proposed KNN_NN hybrid model, first of all we used KNN clasification method for anomaly detection as a binary classification of 'normal' and 'abnormal' classes. After getting the output from the KNN classifier, 'normal' classes are separated and then 'abnormal' classes are used as a input of Neural Network classifier to detect specific attack type as misuse detection. Our hybrid KNN_NN model produced better result than separately used KNN and Neural Network classification method.

VI. Conclusion

The main purpose of this paper is to improve the performance of classifier for intrusion detection. We have achieved it through the use of our proposed hybrid KNN_NN multilevel classifier. Here we test our proposed algorithm on NSL-KDD benchmark dataset. The dataset has been used in three ways for classification. First we use all the 41 features of NSL-KDD dataset for classification. Then classification method is performed on 25 selected features. Selection has been done by Rough Set Theory and Information Gain separately. The result shows that there is a significant decrease in memory space and also decrease in learning time of the algorithm as well as increase in the accuracy. The experimental results emphasized that for NSL-KDD benchmark dataset, Information Gain is the suitable technique as compared to Rough Set Theory for feature selection because of its computational efficiency for large number of features. The experimental result also shows that, our proposed KNN_NN hybrid multilevel classification model is the convenient and effective classification methodology which can be used in the field of intrusion detection.

References

[1] Manthira M. S, Rajeswari M. Virtual Host based Intrusion Detection System for Cloud. International Journal of Engineering and Technology (IJET), Vol.5, No.6, pp.5023-5029(2013).

[2] Araujo J. D, Abdelouahab Z. Virtualization in Intrusion Detection System: A Study on Different Approaches for Cloud Computing Environments. International Journal of Computer Science and Network Security (IJCSNS), Vol.12, No.11, pp.9-16(2012).

[3] Modi C, Patel D, Borisaniya B, Patel H. A survey of intrusion detection techniques in Cloud. Journal of Network and Computer Applications, Vol.36, pp.42-57(2013).

[4] Majeed S. K, Hashem S. H, Gbashi I. K. Propose HMNIDS Hybrid Multilevel Network Intrusion Detection System. International Journal of Computer Science Issues (IJCSI), Vol.10, Issue.5, No.2, pp.200-208(2013).

[5] Garcia-Teodoro P, Diaz-Verdejo J, Macia-Fernandez G, Vazquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, Vol.28, pp.18-28(2009).

[6] Bahrololum M, Salahi E, Khaleghi M. Anomaly Intrusion Detection Design Using Hybrid Of Unsupervised And Supervised Neural Network. International Journal of Computer Networks & Communications (IJCNC), Vol.1, No.2, pp.26-33(2009).

[7] Ghali N. I. Feature Selection for Effective Anomaly-Based Intrusion Detection. International Journal of Computer Science and Network Security (IJCSNS), Vol.9, No.3, pp.285-289(2009).

[8] Datti R, Verma B. Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis. International Journal on Computer Science and Engineering (IJCSE), Vol.2, No.4, pp.1072-1078(2010).

[9] Lakhina S, Joseph S, Verma B. Feature Reduction using Principal Component Analysis for Effective Anomaly–Based Intrusion Detection on NSL-KDD. International Journal of Engineering Science and Technology, Vol.2, pp.1790-1799(2010).

[10] Suguna N, Thanushkodi K. An Improved k-Nearest Neighbor Classification Using Genetic Algorithm. International Journal of Computer Science Issues (IJCSI), Vol.7, Issue 4, No.2, pp.18-21(2010).

[11] Al-Janabi S. T. F, Saeed H. A. A Neural Network Based Anomaly Intrusion Detection System. IEEE Computer Society, pp.221-226(2011).

[12] Sadek R. A, Soliman M. S, Elsayed H. S. Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction. International Journal of Computer Science Issues (IJCSI), Vol.10, Issue 6, No.2, pp.227-233(2013).

[13] Mahmood D. Y, Hussein M. A. Intrusion Detection System Based on K-Star Classifier and Feature Set Reduction. IOSR Journal of Computer Engineering (IOSR-JCE), Vol.15, Issue 5, pp.107-112(2013).

[14] Bhavsar Y. B, Waghmare K. C. Intrusion Detection System Using Data Mining Technique: Support Vector Machine. International Journal of Emerging Technology and Advanced Engineering, Vol.3, Issue 3, pp.581-586(2013).

[15] Nsl-kdd data set for network-based intrusion detection system. Available on: http://nsl.cs.unb.ca/NSL-KDD/, (2009).

[16] Tavallaee M, Bagheri E, Lu W, Ghorbani A. A. A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 53-58(2009).

[17] Ibrahim L. M, Basheer D. T, Mahmod M. S. A Comparison Study For Intrusion Database (KDD99, NSL-KDD) Based On Self Organization Map (SOM) Artificial Neural Network. Journal of Engineering Science and Technology, Vol.8, No.1, pp.107 – 119(2013).

[18] Baig Z. A, Shaheen A. S, Aal R. A. One-Dependence Estimators for Accurate Detection of Anomalous Network Traffic. International Journal for Information Security Research (IJISR), Vol.1, Issue 4, pp.202-210(2011).

[19] Bazan J. G, Szczuka M. The Rough Set Exploration System. Springer-Verlag Berlin Heidelberg, pp.37-56(2005).

[20] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten I. H. The WEKA Data Mining Software: An Update. SIGKDD Explorations. Vol.11, Issue 1, pp. 10-18(2009).

[21] Kohavi R, John G. H. Wrappers for feature subset selection. Artificial Intelligence, pp.273-324(1997).

[22] Han J, Kamber M, Pei J. Data Mining Concepts and Techniques. Elsevier Book, 3rd edition (2012).

[23] Kotsiantis S, Kanellopoulos D. Discretization Techniques: A recent survey. GESTS International Transactions on Computer Science and Engineering, Vol.32, pp.47-58(2006).

[24] Chimphlee W, Abdullah A. H, Md Sap M. N, Chimphlee S, Srinoy S. Unsupervised Clustering Methods for Indentifying Rare Events in Anomaly Detection. International Journal of Computer, Information, Systems and Control Engineering, Vol. 1, No. 8, pp.2578-2583(2007).

[25] Zainal A, Maarof M. A, Shamsuddin S. M. Feature Selection using Rough-DPSO in Anomaly Intrusion Detection. Springer Berlin Heidelberg, pp.512-524(2007).

[26] Pawlak Z. Rough Sets Theoretical Aspect of Reasoning about Data. Kluwer Academic Publishers Book (1991).

[27] Sanchez-Marono N, Alonso-Betanzos A, Tombilla-Sanroman M. Filter Methods for Feature Selection – A Comparative Study. Intelligent Data Engineering and Automated Learning. Springer Berlin Heidelberg, pp.178-187(2007).

[28] Yu L, Liu H. Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, pp.856-863(2003).

[29] Jain M, Richariya V. An Improved Techniques Based on Naïve Bayesian for Attack Detection. International Journal of Emerging Technology and Advanced Engineering, Vol.2, Issue 1, pp.324-331(2012).

[30] Azhagusundari B, Thanamani A. S. Feature Selection based on Information Gain. International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol.2, Issue 2, pp.18-21(2013).

[31] Sethuramalingam S, Naganathan E. R. Hybrid feature selection for network intrusion. International Journal on Computer Science and Engineering. (IJCSE), Vol.3, No.5, pp.1773-1780(2011).

[32] Bishop C. M. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer Book, New York, NY, USA (2007).

[33] Yang L, Jin R. Distance metric learning: a comprehensive survey. Tech. Rep. Department of Computer Science and Engineering, Michigan State University (2006).

[34] Sun N. Q, Li Y. Intrusion Detection Based on Back-Propagation Neural Network and Feature Selection Mechanism. Springer Berlin Heidelberg, pp.151-159 (2009).

[35] Lai L. B, Chang R. I, Kouh J. S. Detecting Network Intrusions Using Signal Processing with Query-Based Sampling Filter. EURASIP Journal on Advances in Signal Processing (2009).

[36] Zhang G. P. Neural Networks for Classification: A Survey. IEEE transaction on systems, man, and cybernetics-part c: applications and reviews, Vol. 30, No. 4, pp.451-462(2000).

[37] Shinghal R. Pattern Recognition Techniques and Applications. Oxford University Press Book (2009).

[38] Demuth H, Beale M. Neural Network Toolbox User's Guide, For Use with MATLAB. Version. 4(2001).