A Hybrid Intrusion Detection System: Integrating Hybrid Feature Selection Approach with Heterogeneous Ensemble of Intelligent Classifiers

Amrita, Kiran Kumar Ravulakollu
(Corresponding author: Amrita)

Department of Computer Science and Engineering, Sharda University
SET, Plot No. 32-34, Knowledge Park III, Greater Noida, Uttar Pradesh 201306, India
(Email: amrita.prasad@sharda.ac.in)
(Received Aug. 9, 2016; revised and accepted Nov. 21 & Dec. 23, 2016)

Abstract

This paper proposes Hybrid Feature Selection Approach – Heterogeneous Ensemble of Intelligent Classifiers (HyFSA-HEIC) for intelligent lightweight network intrusion detection system (NIDS). The purpose is to classify for anomaly from the incoming traffic. This system hierarchically integrates HyFSA and HEIC. The HyFSA will obtain the optimal number of features and then HEIC is built using these optimal features. HyFSA helps to decrease the computation time of the system and make it lightweight to work in real time. The aim of HEIC is to obtain accurate and robust classifier and enhance overall performance of the system. The results demonstrate that proposed system outperforms other ensemble and single classifier methods used in this paper. It has true positive rate (99.9%), accuracy (99.91%), precision (99.9%), receiver operating characteristics (99.9%), low false positive rate (0.1%) and lower root mean square error rate (3.06%) with a minimum number of selected 6 features. It also reduces time to build and time to test the model by 50.79% and 55.30% respectively on reduced features set. The results evince that detection rate, accuracy and precision of the system is increased by incorporating feature selection approach with heterogeneous ensemble of intelligent classifiers and significantly reduce the computation time.

Keywords: Classifier; Ensemble; Feature Selection; Network Intrusion Detection System

1 Introduction

Information security has become an essential key component in all areas with the increasing Internet connectivity and traffic volume. Also the security of networks plays a vital role in information security [27]. Therefore, Intrusion Detection Systems (IDS) for network have become an essential component for information security to protect it from continuous increase of network-based intrusions and attacks. The role of Network Intrusion Detection Systems (NIDS) is to actively monitor the function of network and detect malicious activities in real time and raise an alert. Network intrusion detection is a classification task that is capable of distinguishing between normal and attack or intrusion traffic connection i.e. two-class problem and further classification of the different attacks type. Also NIDS has to examine huge amount of data with high dimensional network traffic in real time and on-line processing. Therefore, it is necessary to build accurate, intelligent and lightweight NIDS to protect networks as well as information system.

In high dimensional feature set, some features may be redundant and some others may be irrelevant. These redundant and irrelevant features can increase the computation time and degrade the performance and accuracy of NIDS. For this reason feature selection method is used as a pre-processing step to obtain the subset of relevant features to construct NIDS. It selects the minimal cardinality feature subset that maintains the detection rate and accuracy as the original feature set. For classification, the main issue is to select the best classification method as every method has its own advantages and disadvantages [11]. Therefore heterogeneous ensemble of intelligent classifiers has been proposed to overcome the limitation of single classification method. Ensemble method exploits the strength of each classifier of the ensemble to acquire accurate and robust classifier. It combines different classification method to improve the performance and accuracy of the model.

The ultimate goal of NIDS is to achieve best possible detection accuracy and reliability. It can be achieved by

combining different decision-making model into one system. This leads to the design of hybrid system. The hybrid system integrates different decision-making models or learning techniques to boost the performance of the system than the individual decision making or learning technique. The primarily focus on the design of hybrid system is integration and interaction of different learning techniques covering computational phases from data preprocessing up to final decision making.

In an attempt to develop lightweight and efficient anomaly based NIDS for two-class classifications, i.e., intrusions or attack and normal, a novel hybrid system for network intrusion detection, HyFSA-HEIC (Hybrid Feature Selection Approach – Heterogeneous Ensemble of Intelligent Classifiers) is proposed. The HyFSA-HEIC integrates hybrid feature selection approach (HyFSA) and heterogeneous ensemble of intelligent classifiers (HEIC) to make it lightweight and accurate. HyFSA proposed in [3] has been used for the selection of optimal features set. To increase the performance of NIDS, HEIC has been developed. The performance of HyFSA-HEIC has been tested on non-redundant datasets of "10% KDD" and "Corrected Test". True Positive Rate (TPR), Precision (PRE), Accuracy (ACC), False Positive Rate (FPR), Receiver Operating Characteristics (ROC), Time-span to build model (TBM), Time-span to test model (TTM) and Root mean squared error (RMSE) have been used as performance evaluation metrics.

The paper is organized as follows. Review of related work is presented in Section 2. Section 3 introduces feature selection approaches, classification methods and ensembles employed in this work. Performance evaluation measures utilized in this paper are discussed in Section 4. The proposed system and experimental setup adopted in this paper is presented in Section 5, experimental results and analysis in Section 6, and conclusion and future direction in Section 7.

2 Related Works

Many hybrid based systems or approaches have been proposed and investigated in the literature to enhance the accuracy of the IDS in the recent years. Each technique has its own strength and weakness. As well as performance of each technique is varies in terms of Detection Rate Accuracy, Precision, False Positive Rate and error rate. Panda et al. [23] investigated novel hybrid intelligent technologies for making intelligent decision which combines the supervised or unsupervised with classifier using data filtering to detect network attacks. The approach was evaluated on NSL-KDD dataset for 2-class classification and has 99.9% detection rate with 0.06% error. Agarwal et al. [1] presented a hybrid approach which combines entropy and Support Vector Machine (SVM) for anomaly network traffic detection system. The Hybrid method outperforms the single method in terms of accuracy with 97.25% and misclassified instances of 2.75%. Chitrakar et al. [7] proposed two hybrid approaches for anomaly intrusion detection. In first approach, k-medoids clustering is combined with classifier Naïve Bayes and in second, k-medoids clustering is combined with classifier Support Vector Machine. These approaches show enhancement in the TPR and reduction in the FPR.

Sindhu et al. [26] proposed a lightweight Network IDS employing a wrapper based feature selection approach with neural ensemble of decision trees to maximize the specificity and sensitivity. The average classification rate and error of the proposed system with 16 selected features is 98.4% and 1.62% and performed better than C4.5, Naïve Bayes, Decision Stump, REP tree, Random Tree and Random Forest. A reliable and efficient IDS based on gradually feature removal method with the combination of k-mean, ant colony algorithm and support vector machine has been developed by Li et al. [20] for normal or attack detection in network. The system was evaluated on KDD Cup 99 data set. Nineteen features were selected by applying gradually feature removal method with accuracy of 98.62%. Lin et al. [21] combined support vector machine, decision tree and simulated annealing for anomaly based intrusion detection. In this, support vector machine and simulated annealing were used to obtain the best selected features using KDD dataset and decision tree and simulated annealing were used to find decision rules for new attacks which can enhance the accuracy of the method. The performance of the proposed algorithm outperforms other existing approaches with weighted average TPR and FPR of 98.3% and 1.4% respectively.

A new hybrid intrusion detection method which hierarchically integrates a misuse detection model and an anomaly detection model was proposed by Kim et al. [16]. The C4.5 decision tree algorithm has been used to build the misuse detection model. This model is then used to decompose normal training data into smaller subsets. The one-class support vector machine (SVM) is used to build the anomaly detection model for each decomposed subset. NSL-KDD dataset has been used to evaluate the proposed method. The experimental results demonstrated that method was better in term of detection rate for both known and unknown attacks and reduced the training and testing time of the model. A hybrid approach to anomaly detection using a real-valued negative selection based detector generation in the large scale dataset is presented in [13]. It uses k-mean clustering to reduce the size of the training dataset to identify good starting points for the detector generation based on multi-start metaheuristic method and genetic algorithm. The results showed that this approach outperforms other techniques by 96.1% accuracy with time of 152 s and low false positive rate of 0.033. Vahid Golmah [14] developed a hybrid method to improve the accuracy of the intrusion detection system based on C5.0 and SVM. The proposed method is evaluated on benchmark "KDD Cup 1999" dataset with full feature set. The average precision of classification for the proposed algorithm is 99.96%.

3 Related Background

3.1 Feature Selection Method

Feature selection method is commonly used to find the optimal feature subset to improve the system performance by eliminating redundant and irrelevant features from dataset. It also helps to alleviate "the curse of dimensionality". There are three approaches—*filter, wrapper* and *hybrid* for feature selection [4]. Filter approach [22] utilizes external classifier to assess the performance of selected features. The wrapper approach [17] "wrap around" the predetermined classifier to assess subset of features. This method is computationally more expensive than the filter method [9, 17]. The hybrid approach [9] combines the filter and wrapper approach to achieve the best possible performance with a specific classifier. In this work, HyFSA [3] has been used for the selection of optimal features set. A survey of several works on feature selection approaches applied on "KDD Cup 1999" dataset for IDS is presented in [2].

3.2 Classification Methods

The base classifiers selected for the ensemble are probability theory based Naïve Bayes (NB) [31], decision tree based C4.5 [24], homogeneous ensemble of decision trees based random forest (RF) [6], soft computing based Neural networks using Stochastic Gradient Descent (NN-SGD) [5], instance based k-Nearest Neighbor (kNN) [28], and rule based Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [8]. These base classifiers are briefly discussed below.

3.2.1 Naïve Bayes Classifier

Naïve Bayes (NB) classifier [31] is a supervised classifier. It is based on Bayes' theorem. This classifier computes the posterior probability for each class c_j to classify an input pattern $\vec{x}_i$ and assigns the target class c* with the highest posterior probability to $\vec{x}_i$ using Equation (2). The output of the individual classifiers as a posteriori probability can be represented as $P(c_j|\vec{x}_i)$, where $\vec{x}_i$ is presented to i^{th} classifier and assigned to class c_j. For two-class classification, the posteriori probability using Bayes theorem can be calculated as

$$P(c_j|\vec{x}_i) = \frac{P(\vec{x}_i|c_j)P(c_j)}{P(\vec{x}_i)} = \frac{P(\vec{x}_i|c_j)P(c_j)}{P(\vec{x}_i|c_1)P(c_1) + P(\vec{x}_i|c_2)P(c_2)} \quad (1)$$

where j=1,2;i=1,...,L

$$c^* = arg\ max P(c_j|\vec{x}_i) \quad (2)$$

where $P(c_j|\vec{x}_i), P(c_j), P(\vec{x}_i|c_j)$ *and* $P(\vec{x}_i)$ are called the posterior probability, prior probability, likelihood, and evidence respectively. Naïve Bayes classifier can work on symbolic as well as numerical features. It exhibits high speed and accuracy and highly suitable for high dimensional large dataset [31].

3.2.2 Decision Tree

Decision tree (DT) is a supervised learning algorithm based on tree-like structure which consists of nodes and branches. Each non-terminal node represents a test on an attribute, each branch represents the outcome of the test and each leaf node represents the class label for classification of the input pattern. The classification of input pattern starts from the root node and follow the branch to reach the leaf node of the DT. A well-known algorithm for constructing decision tree is C4.5 [24]. The C4.5 algorithm is also very robust for high dimensional data and handling missing data. It works well on both numerical and symbolic features.

3.2.3 Random Forest

Random Forest (RF) [6] is an ensemble based classification techniques. It generates many unpruned decision tree by inducing different bootstrap sample using random feature selection from training dataset. It is called random forest as sampling of records is done randomly and forest of decision trees are built in the process. Final class of an input instance is made by aggregating the decisions of the individual trees in the forest by majority voting for classification. Random forest works efficiently on high dimensional large dataset and able to deal with unbalanced and missing data.

3.2.4 Neural Network

Neural network or Artificial Neural network (ANN) is computational technique that mimics the neurons of human brain. It consists of set of simple processing components called artificial neurons that are interconnected to other neuron by synapses or link. These neurons are organized into network in many ways known as topologies. The neurons in ANN are grouped into layers as input, output and hidden layer. Each link is associated with weight. Neurons in the input layer receive stimuli from outside the network, transform it into output and pass the output to the subsequent hidden or output layer. The network learns by adjusting the corresponding weight of the link in the learning phase. It helps ANN to assign the correct class label to the given input pattern. Most commonly used ANN architecture is Stochastic Gradient Descent (SGD) [5] for large scale dataset and online learning.

3.2.5 K-Nearest Neighbor

K-nearest neighbor (kNN) [28] is supervised classification method. It is simple, non-parametric, instance-based and lazy learning algorithm. Lazy learning signifies that the algorithm does not build the model until the time classification is required. This algorithm classifies the new input

pattern by calculating the similarity measure (e.g. distance function) between the new input pattern and each instance of training dataset and then uses the class labels of the k most similar neighbors to assign the class of new input pattern based on majority voting. Euclidean distance or the cosign value can be used as similarity measures to calculate the similarity between two instances. The performance of this classifier relies on the value of k.

3.2.6 RIPPER

Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [8] is supervised rule-based learning algorithm. This is an extension of IREP (Incremental Reduced Error Pruning). It improved the efficiency of IREP by reducing errors by applying repeated pruning, faster training time, support missing attributes and noisy datasets. This learning algorithm searches the feature set of the training dataset and produces concise rule-sets for each class label. It works efficiently on numerical and large dataset.

3.3 Ensemble of Classifiers

An ensemble of classifiers combines multiple weak or diverse classifiers whose individual outputs are combined in some means to form a final decision [10]. The combined decisions of an ensemble generally provide better performance than the individual classifiers [25]. The main motivation of using ensemble of classifiers is to get better accuracy of the complex problem by exploiting the strengths of individual classifiers with the aim to obtain the best possible collective decision accuracy than any of the individual classifiers. Ensemble of multiple diverse classifiers has more reliable and better decision than single classifier as it reduces the chance of incorrect classification done by single classifier and also overcome the limitation of single classifier. The architectures of the ensemble of classifiers are mainly categorized into two types, i.e., parallel and serial. There are two steps for constructing an ensemble: (1) generating the base learning algorithms or classifiers, and (2) combining the decisions of base learning algorithms for maximum accuracy.

3.3.1 Generating Base Classifiers

In this step, individual classifier of the ensemble known as base classifier is generated. Methods for generating ensemble can be categorized as Homogenous and Heterogeneous ensemble. Homogeneous ensemble can be generated from the different executions of the same classifier. This ensemble can be generated by using any method from (i) different subset of training data with same classifier, (ii) different set of input training parameters available with a single classifier, (iii) different feature sets, (iv) multi-class specialized systems, or (v) manipulation of output labels. Examples are Bagging, Boosting, Option Trees, Error-correcting output codes. Heterogeneous ensemble uses different learning algorithm or classifiers on the same data set. Different methods for generating heterogeneous ensemble are: (i) Voting or fixed-rule aggregation, and (ii) Stacked generalization or meta-learning.

The classifiers in ensemble should be accurate and diverse to improve the performance of ensemble system over single classifier. Therefore, classifiers must be highly accurate and diverse to build efficient and accurate ensemble [19]. Weak classifiers in the ensemble also result in weak ensemble and hence deteriorate the accuracy of ensemble. The classifiers are said to be diverse or unique if they make distinct errors on distinct instances and combining their outputs can decrease the total error. Also diverse classifiers contribute toward uncorrelated decisions and improve the overall accuracy of the ensemble system. There are many methods to achieve classifier diversity [18]. Any method of homogeneous or heterogeneous described in section 3.3.1 can be used to generate diversity among the classifiers.

3.3.2 Combining Classifiers

The second step in ensemble method is the approach employed in combining the decision of the classifiers. The main motivation for combining multiple classifiers is to obtain a consensus decision by combining the individual decision of classifiers [18]. There are mainly two approaches in combining the decisions of different classifiers—*classifier selection* and *classifier fusion* [18]. The classifier selection approach selects a single classifier to give the final decision for a new instance while classifier fusion approach combines the decision of all classifiers. The various combination methods have been reported in [18]. The most commonly used method is elementary combiners based on algebraic combination rules. This method combines the decisions of classifiers that can be expressed as a posteriori probability. The major benefit of using this method is its simplicity as it does not need any training. It includes several methods as *Sum, Average, Product, Majority Voting, Minimum*, and *Maximum* rules.

Let $\{D_1, D_2, ..., D_L\}$ be the set of L individual classifiers and $\{c_1, c_2, ..., c_m\}$ be the set of m possible class labels. The combiner combines the decisions of all D_i to predict the final class label for the input instance $\vec{x}_i$. In order to employ the *Sum, Average, Product, Majority Voting, Minimum*, and *Maximum* methods, outputs of all D_i can be viewed as a posteriori probabilities using Bayes theorem defined in Equation (1). Let input instance $\vec{x}_i$ is finally assigned to class c, where c is one of the m possible classes. The *Sum, Average, Product, Majority Voting, Minimum,* and *Maximum* methods can be used to determine c are defined as follows:

$$\text{Sum Rule: } c = max_{j=1...m} \sum_{i=1}^{L} P(c_j|\vec{x}_i)$$

$$\text{Average Rule: } c = max_{j=1...m} \frac{1}{\mathrm{L}} \sum_{i=1}^{L} P(c_j|\vec{x}_i)$$

$$\text{Product Rule: } c = max_{j=1...m} \prod_{i=1}^{L} P(c_j|\vec{x}_i)$$

$$\text{Majority Voting Rule: } c = max_{j=1...m} \sum_{i=1}^{L} \Delta_{ji} \quad (3)$$

$$\text{Minimum Rule: } c = max_{j=1..m} min_{i=1..L} P(c_j|\vec{x}_i)$$

$$\text{Maximum Rule: } c = max_{j=1..m} max_{i=1..L} P(c_j|\vec{x}_i)$$

In the decision rule in Equation (3), $\Delta_{ji} = 1\ if\ P(c_j|\vec{x}_i) = max_{j=1..m} P(c_j|\vec{x}_i)$ and zero otherwise.

4 Performance Evaluation Measures

Several performance evaluation methods are employed to assess the accuracy and efficiency of the HyFSA-HEIC, classifiers and for comparison. The performance evaluation methods are *True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), Receiver Operating Characteristics (ROC) or Area Under Curve (AUC), Time-span to Build the Model (TBM), Time-span to Test the Model (TTM)* and as follows:

$$True\ positive\ rate(TPR) or\ Recall(R)\ or$$

$$Sensitivity\ or\ Detection\ rate(DR) = \frac{TP}{TP+FN}$$

$$False\ positive\ rate(FPR) = \frac{FP}{FP+TN}$$

$$Accuracy\ (ACC) = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision\ (PRE) = \frac{TP}{TP+FP}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (P_i - T_i)^2} \quad (4)$$

where T_i is the true value, P_i is the prediction, and N is the number of observations in Equation (4).

5 Proposed System and Experimental Setup

The aim of this paper is to propose a Hybrid Feature Selection Approach – Heterogeneous Ensemble of Intelligent Classifiers (HyFSA-HEIC), a hybrid system for network intrusion detection for the classification of coming input pattern into either normal or intrusion. This system must be accurate, lightweight, low false positive rate, high detection rate and able to work in real time. The HyFSA-HEIC integrates the hybrid feature selection approach (HyFSA) with heterogeneous ensemble of intelligent classifiers (HEIC). This is a hierarchical system in which HyFSA selects the optimal feature set for the classification of normal or attack pattern. Then selected optimal feature set is provided as an input to next layer i.e. HEIC for final decision. The overall accuracy of the system relies on the accurate functionality of all layers in the system. Weka 3.7.13 [15] is used as tool for classifiers, feature selection approaches and ensembles utilized in this paper. Figure 1 illustrates the architecture of the HyFSA-HEIC. It contains following five phases:

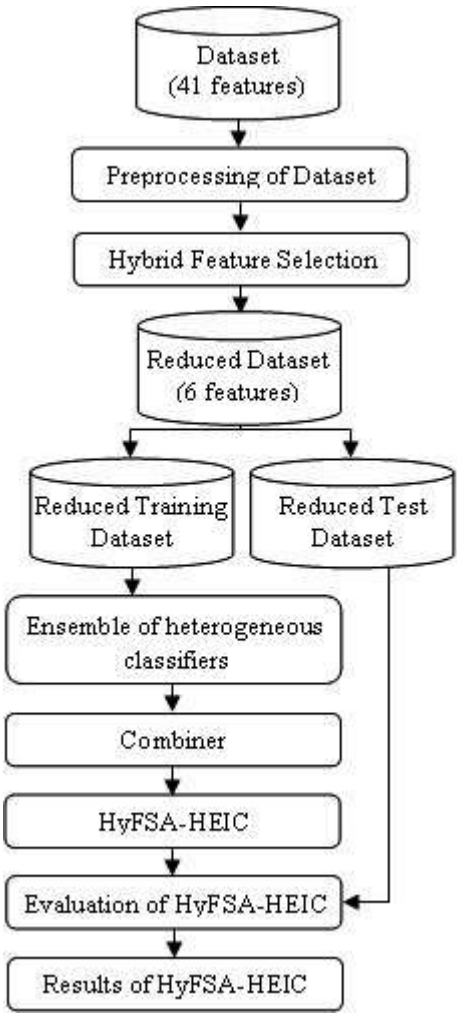


Figure 1: Framework of proposed HyFSA-HEIC

Phase I: Construction of dataset;

Phase II: Feature Selection;

Phase III: Selection of base classifiers for ensemble;

Phase IV: Ensemble and combiner method;

Phase V: Evaluation of classifiers and ensembles.

5.1 Phase I: Construction of Dataset

The "KDD Cup 1999" dataset [29] is the benchmark dataset for intrusion detection and derived from DARPA 1998 dataset. It is the most widely used comprehensive dataset used by many researchers for NIDS. "KDD

Cup 1999" is comprised of three independent datasets—"Whole KDD", "10% KDD" and "Corrected Test". The "10% KDD" has 494,021 connection records in which 97277 are normal and 396744 are attack whereas "Corrected Test" has 311,029 connection records in which 60,593 are normal and 250,436 are attacks shown in Table 1. Each connection record has 41 features (32 numerical and 9 categorical) numbered in an order of 1,2,3,4....,41 plus one class label. Each connection record has label of either normal or attack, with a specific kind of attack type. The attacks fall into one of the four main classes—Probe, User to Root (U2R), Remote to Local (R2L), and Denial of Service (DoS). The "Whole KDD" dataset consists of 22 attack types and "Corrected Test" dataset includes 17 additional attack types and hence total 39 attack types.

NSL-KDD dataset [30] is another widely used dataset for anomaly detection in NIDS. This dataset is refined version of "KDD Cup 1999" dataset. It consists of selected and non-redundant connection records with same number of features of "KDD Cup 1999" dataset. Data cleaning step of data preprocessing can be skipped by using this dataset. Data preprocessing is an important step in any decision making system. Therefore "KDD Cup 1999" dataset has been used as an experimental dataset to make a complete system from preprocessing step to final decision making step and more suitable for real time and on-line processing.

The "10% KDD" and "Corrected Test" datasets are selected as experimental dataset and preprocessed for binary class classification (normal or attack). This phase contains 4 steps. **1) Data transforming:** For binary class classification, the label must have either normal or attack for each connection. Therefore label of each connection for all types of attack are transformed into label "attack". **2) Removal of redundant records:** The converted binary class dataset also consists of large number of redundant connection records. These redundant records will cause classifiers to be influenced towards redundant records in the training dataset. It will also influence the performance of the learning algorithms. The "10% KDD" and "Corrected Test" datasets contain around 70% and 75% redundant records respectively in which the attack class has most of the redundant records than normal class. The resultant datasets are named as "Unique 10% KDD" and "Uni Corr Test". **3) Discretization of dataset:** Feature selection approaches employed in this paper work on discrete data and "10% KDD" dataset has 32 numerical features. Therefore, discretization approach presented by [12] based on Entropy Minimization is utilized. The "Dis_Unique 10% KDD" is resultant discretized dataset. **4) Construction of training and test dataset:** "Unique 10% KDD" is equally partitioned into two datasets: the training dataset ("Uni Train") and test dataset ("Uni Test") for all 41 features. Each dataset contains 72793 records in which each class comprises 50% of the data of "Unique 10% KDD". Reduced training dataset ("Red Uni Train") and reduced test dataset ("Red Uni Test") for selected 6 features in phase II are created from "Uni Train" and "Uni Test" datasets respectively. "Uni Corr Test" and "Reduced Uni Corr Test" are also employed as another test datasets for all 41 and reduced 6 features respectively. Training datasets are utilized to build and test datasets are utilized to assess the performance of the classifiers and ensembles. Table 1 illustrates the statistics of the records for normal and attack in "10% KDD", "Unique 10% KDD", "Corrected Test" and "Uni Corr Test" datasets respectively.

5.2 Phase II: Feature Selection

The accuracy and efficiency of the IDS also depends on the dimension of the dataset. Hybrid method for feature selection proposed in [3] has been used to obtain the optimal number of features for binary (normal or attack) classification. This method employs fusion of filter based feature selection approaches and wrapper method using Naïve Bayes classifier. Filter based feature selection methods used for fusion are consistency-based feature selection (CON), gain ratio (GR), correlation-based feature selection (CFS) and information gain (IG). First, the common features are selected from the feature subsets obtained by applying CFS and CON with best first search. Similarly, the common features are selected from the feature subsets obtained by applying IG and GR. Then, initial feature subset is created by adding these two common feature subsets. Another left feature set is created by adding the remaining features from the sets of CFS, CON, IG and GR. Wrapper based feature selection method employed Naïve Bayes classifier is further applied to obtain the final optimal feature subset. Linear Forward selection (LFS) is used in wrapper method. It starts with the initial feature subset and then add feature one by one from the left feature set until there is no change in the performance of current feature subset. This feature selection approach selects the relevant features and removed redundant and irrelevant features. Finally, 6 best features are selected from 41 features by employing Hybrid feature selection approach [3] on "Dis_Unique 10% KDD" dataset consisting of 41 features. These feature's name and number are {Service-3, Src-bytes-5, Dst-bytes-6, Hot-10, Num-compromised-13, Same-srv-rate-29}. For detail procedure for feature selection and performance of selected features, refer [3].

5.3 Phase III: Selection of Base Classifiers for Ensemble

In HyFSA-HEIC, a novel heterogeneous ensemble is presented that combines the decisions of diverse and accurate learning algorithms or classifiers to the problem of normal or attack detection in IDS. The main issues in the ensemble technique are accuracy and diversity of individual classifiers in ensemble. Different learning algorithms or classifiers for constructing an ensemble enforce a high level of diversity [18]. This ensemble has benefit of differ-

Table 1: Instances and percentages of division of normal and attack in "10% KDD", "Unique 10% KDD", "Corrected Test" and "Uni Corr Test" datasets for all 41 and 6 features.

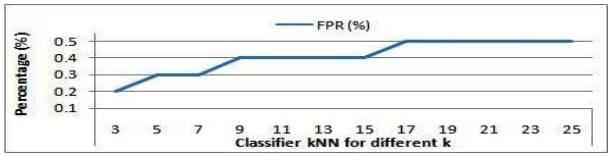
	10% KDD		Unique 10% KDD		Corrected Test		Uni Corr Test	
Type	**#Instance**	**(%)**	**#Instance**	**(%)**	**#Instance**	**(%)**	**#Instance**	**(%)**
Normal	97277	19.69	87832	60.33	60,593	19.48	47,913	61.99
Attack	396744	80.31	57754	39.67	250,436	80.52	29,373	38.01
Total	494021	100	145,586	100	311,029	100	77,286	100

ent biases for each individual classifier and also reduces the bias that can be occurred in a single learning algorithm or classifier. Also combining weak diverse classifiers in the ensemble will result in weak ensemble and lead to deteriorate the accuracy of the ensemble. Therefore, different classifiers are compared to select accurate and diverse base classifiers. The base classifiers selected for the ensemble are C4.5, Naïve Bayes (NB), Neural networks using mini-batch stochastic gradient descent (NN-SGD), K-nearest neighbor (kNN), Repeated Incremental Pruning to Produce Error Reduction (RIPPER) and random forest (RF). Motivation of selecting these different types of base classifiers leads diversity in creating ensemble classifiers. Each selected classifier has different learning hypotheses (trees, instance-based, rules and statistics) and also different inductive bias that make diverse set of classifiers for ensemble. Different learning hypotheses and inductive bias generates diversity among the classifiers. The base classifiers are briefly discussed in Section 3.2.

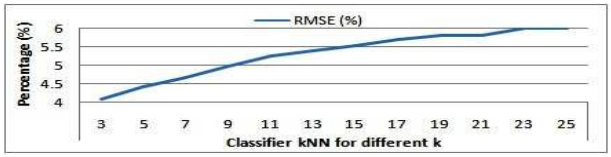
The performance of kNN classifier is influenced by the suitable selection of optimal value of parameter k. In order to select the optimal value of k for kNN classifier, 10-fold cross validation is performed on "Red Uni Train" dataset for 6 features. The value of k is varied from 1 to 25 for odd number for two-class classification to avoid tied votes. The empirical results are shown in Table 2. The value of k for kNN is selected based on the value of k which minimizes errors and maximizes predictive accuracy of the kNN classifier. As can be seen in Table 2, the kNN for $k = 3$ outperformed among all k except $k = 1$. The kNN for $k = 1$ cannot be considered in the selection of k as it badly overfits the classifier. Therefore, three nearest neighbours ($k = 3$) is selected for each instance in kNN classifier. Figure 2 shows performance comparison of kNN for different values of k in terms of FPR, RMSE and ACC respectively. Euclidean Distance as similarity measure has been used to find the nearest neighbours. Euclidean distance $d(u, v)$ between two instances u and v is defined as $d(u, v) = \sqrt{\sum_{i=1}^{N} \delta_i^2}$, where δ_i is the difference between the i^{th} feature's value of the instances u and v and N is the dimension of the dataset. The difference δ_i can be measured for numerical as well as for nominal feature as

$$\delta_i = \begin{cases} x_i - y_i, & for\ numerical\ feature \\ 0, & if\ x_i = y_i\ for\ nominal\ feature \\ 1, & if\ x_i \neq y_i\ for\ nominal\ feature \end{cases} \quad (5)$$

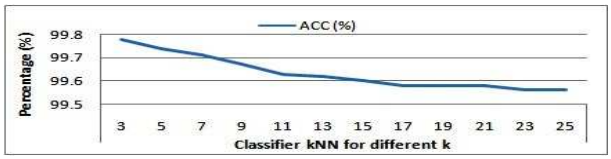
These classifiers are trained on "Uni Train" training dataset. Performance metrics used in the comparisons are TPR, FPR, ACC, PRE, ROC, TBM, TTM and RMSE. The results of these classifiers using 6 selected features and all 41 features based on different performance metrics on training datasets ("Red Uni Train" and "Uni Train") are depicted in Table 3.



(a) FPR



(b) RMSE



(c) ACC

Figure 2: Performance comparison of kNN for different k in terms of (a) FPR, (b) RMSE and (c) ACC

5.4 Phase IV: Ensemble and Combiner Method

A heterogeneous ensemble of classifiers is a collection of multiple diverse classifiers. In this, decision of individual classifiers is combined to classify input instance. A heterogeneous ensemble of classifiers will combine the strength and disagreement of all diverse classifiers and also make individual classifier disagree with each other. The strength and disagreement among the diverse classifiers are utilized by elementary combiners based on algebraic combination rules to give accurate and reliable final decision. To construct the heterogeneous ensemble, a parallel ensemble structure is employed in which each classifier is trained on "Red Uni Train" training dataset independently. Then elementary combiners based on al-

Table 2: Performance of kNN on "Red Uni Training" training dataset for 6 features using 10-fold cross validation

Evaluation Metrics	kNN Classifiers												
	k=1	k=3	k=5	k=7	k=9	k=11	k=13	k=15	k=17	k=19	k=21	k=23	k=25
TPR (%)	99.8	99.8	99.7	99.7	99.7	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6
FPR (%)	0.2	0.2	0.3	0.3	0.4	0.4	0.4	0.4	0.5	0.5	0.5	0.5	0.5
ACC (%)	99.84	99.78	99.74	99.71	99.67	99.63	99.62	99.6	99.58	99.58	99.58	99.56	99.56
PRE (%)	99.8	99.8	99.7	99.7	99.7	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6
ROC (%)	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
RMSE (%)	3.88	4.09	4.42	4.68	4.98	5.26	5.39	5.53	5.68	5.8	5.8	5.99	5.99

Table 3: Performance of classifiers on "Uni Train" and "Red Uni Train" training datasets for 41 and 6 features

Evaluation Metrics	# Features	Classifiers					
		NB	NN-SGD	kNN(k=3)	RIPPER	C4.5	RF
TPR (%)	41	97.0	99.5	99.9	99.9	99.9	99.9
	6	95.1	97.2	99.9	99.8	99.9	99.9
FPR (%)	41	3.8	0.5	0.1	0.2	0.1	0.1
	6	6.1	3.7	0.1	0.2	0.2	0.1
ACC (%)	41	96.99	99.47	99.87	99.85	99.91	99.93
	6	95.12	97.16	99.87	99.83	99.88	99.9
PRE (%)	41	97.0	99.5	99.9	99.9	99.9	99.9
	6	95.2	97.2	99.9	99.8	99.9	99.9
ROC (%)	41	97.6	99.5	100	99.9	100.0	100.0
	6	99.2	96.7	100	99.8	100.0	100.0
TBM (sec)	41	2.56	541.43	0.08	171.51	40.34	952.99
	6	0.45	170.56	0.08	46.28	3.24	38.11
TTM (sec)	41	6.74	29.87	15781.1	0.64	0.58	146.05
	6	1.61	1.61	5087.73	0.21	0.35	8.85
RMSE (%)	41	17.28	7.25	2.84	3.78	2.9	2.56
	6	21.97	16.86	3.09	4.05	3.32	2.85

gebraic combination rules are utilized to fuse the decisions of these base classifiers to produce final decision. In this paper, *Average, Product, Majority Voting, Minimum,* and *Maximum* methods of *classifier fusion* approach are used to fuse the decisions of the classifiers to produce final decision. The reason for using these methods is to achieve good results, very fast computation and their simplicity. Additionally, the heterogeneous ensemble is also constructed using "Uni Train" training dataset (41 features) for comparison. The 5 base classifiers—NB, NN-SGD, RIPPER, C4.5 and RF out of 6 are utilized for the construction of ensemble. The results of these ensembles using reduced 6 and all 41 features based on different performance metrics on training datasets ("Red Uni Train" and "Uni Train") are illustrated in Table 4.

5.5 Phase V: Evaluation of the Classifiers and Ensembles

Datasets "Uni Test", "Red Uni Test", "Uni Corr Test" and "Red Uni Corr Test" have been used to test the effectiveness of classifiers and ensembles used in this paper. Performance metrics were used in the experiments are TPR, FPR, ACC, PRE, ROC, TBM, TTM and RMSE. The performance of these classifiers using all 41 features and 6 selected features based on different performance metrics were evaluated on test datasets ("Uni Test" and "Red Uni Test") are shown in Table 5 and on "Uni Corr Test" and "Red Uni Corr Test" in Table 7. The performance of constructed heterogeneous ensembles using *Average, Product, Majority Voting, Minimum,* and *Maximum* using all 41 features and 6 selected features based on different performance metrics were evaluated on test dataset ("Uni Test" and "Red Uni Test") are illustrated in Table 6 and on "Uni Corr Test" and "Red Uni Corr Test" in Table 8. The classifiers and ensembles were also evaluated on training datasets ("Uni Train" and "Red Uni Train") for 41 and 6 features using 10-fold cross validation. Table 9 illustrates the results of NB, NN-SGD, RIPPER, C4.5, RF and HEIC on training dataset ("Red Uni Train") for 6 features using 10-fold cross validation.

6 Experimental Results and Analysis

To evaluate the performance of HyFSA-HEIC proposed in Section 5 in terms of accuracy and efficiency, several

Table 4: Performance of ensembles on "Uni Train" and "Red Uni Train" training datasets for 41 and 6 features

Evaluation Metrics	# Features	Ensemble of Classifier (Vote)				
		Average	Product	Majority Voting	Minimum	Maximum
TPR (%)	41	100.0	99.8	100.0	99.8	98.1
	6	99.9	99.6	99.9	99.6	97.8
FPR (%)	41	0.0	0.2	0.0	0.2	2.8
	6	0.1	0.5	0.1	0.5	3.2
ACC (%)	41	99.97	99.47	99.97	99.47	98.15
	6	99.9	97.16	99.91	97.16	97.85
PRE (%)	41	100.0	99.8	100	99.8	98.2
	6	99.9	99.6	99.9	99.6	97.9
ROC (%)	41	100.0	99.7	100.0	99.7	100.0
	6	100.0	98.3	99.9	98.3	99.9
TBM (sec)	41	626.23	590.15	541.61	616.26	599.57
	6	227.6	227.12	226.54	264.01	253.69
TTM (sec)	41	21.5	17.86	20.34	21.08	18.08
	6	8.42	9.91	9.09	10.39	10.47
RMSE (%)	41	4.29	3.91	1.85	3.91	9.17
	6	7.49	6.14	3.06	6.14	11.59

Table 5: Performance of classifiers on "Uni Test" and "Red Uni Test" test datasets for 41 and 6 features

Evaluation Metrics	# Features	Classifiers					
		NB	NN-SGD	kNN(k=3)	RIPPER	C4.5	RF
TPR (%)	41	97.1	99.5	99.8	99.9	99.9	100.0
	6	95.2	97.2	99.9	99.9	99.8	99.9
FPR (%)	41	3.7	0.5	0.2	0.1	0.1	0.1
	6	6.1	3.6	0.2	0.2	0.2	0.1
ACC (%)	41	97.06	99.48	99.81	99.92	99.87	99.96
	6	95.17	97.23	99.84	99.86	99.85	99.92
PRE (%)	41	97.1	99.5	99.8	99.9	99.9	100.0
	6	95.2	97.3	99.8	99.9	99.8	99.9
ROC (%)	41	97.6	99.5	99.9	99.9	99.9	100.0
	6	99.3	96.8	100.0	99.9	99.9	100.0
RMSE (%)	41	17.07	7.24	3.94	2.81	3.33	2.15
	6	21.87	16.64	3.67	3.62	3.63	2.54

experiments have been performed. All experiments were conducted on an Intel (R) @ 2.13 GHz Core (TM) i3 CPU M 330 computer with 2.87 GB memory and Windows 7 Home Premium operating system in Java Environments Weka 3.7.13. Datasets were used in the experiments for training are "Uni Train" and "Red Uni Train", and for testing are "Uni Test", "Red Uni Test", "Uni Corr Test" and "Red Uni Corr Test" depicted in Table 1.

Firstly, hybrid feature selection approach (Phase II, Section 5) was utilized to obtain the optimal features set for the identification of normal or intrusion instances. These features set obtained based on performance is reduced to 15% from 41 to 6 features. Then training and testing were performed for 6 diverse classifiers and 5 ensembles built from these classifiers using reduced 6 and all 41 features sets and then compared these models using 6 features with those using 41 features on different evaluation metrics.

From Table 3, as can be seen, all 6 classifiers obtained same or better performance on original 41 features than reduced 6 features set on all evaluation metrics except TBM and TTM. It is evident that classifier RF is equally best on 41 features as well as on 6 features set with TPR(99.9%), FPR (0.1%), PRE (99.9%), and ROC (100.0%) and also outperforms other five classifiers in terms of all evaluation metrics but TBM and TTM. This classifier has ACC of 99.93% and 99.90%, and RMSE of 2.56% and 2.85% on 41 and 6 features respectively. The classifier kNN has minimum TBM of 0.08 sec because no explicit training step is required. Apart from this, among all classifiers, NB has minimum TBM of 2.56 sec and 0.45 sec on 41 and 6 features respectively, C4.5 has minimum TTM of 0.58 sec on 41 features, RIPPER has minimum TTM of 0.21 sec on 6 features as depicted in Table 3. The performance of the classifier kNN is same on both 41 and 6 features set in terms of TRP (99.9%), FPR (0.1%), ACC

Table 6: Performance of ensembles on "Uni Test" and "Red Uni Test" test datasets for 41 and 6 features

Evaluation Metrics	# Features	Ensemble of Classifier (Vote)				
		Average	Product	Majority Voting	Minimum	Maximum
TPR (%)	41	99.9	99.8	99.9	99.8	98.1
	6	99.9	99.6	99.9	99.6	97.8
FPR (%)	41	0.1	0.2	0.1	0.2	2.8
	6	0.2	0.5	0.2	0.5	3.2
ACC (%)	41	99.94	99.45	99.94	99.45	98.15
	6	99.87	97.22	99.88	97.22	97.84
PRE (%)	41	99.9	99.8	99.9	99.8	98.2
	6	99.9	99.6	99.9	99.6	97.9
ROC (%)	41	100.0	99.6	99.9	99.6	100.0
	6	100.0	98.4	99.9	98.4	99.9
RMSE (%)	41	4.44	4.44	2.49	4.44	9.19
	6	7.5	6.27	3.42	6.27	11.6

Table 7: Performance of classifiers on "Uni Corr Test" and "Red Uni Corr Test" test datasets using 41 and 6 features

Evaluation Metrics	# Features	Classifiers					
		NB	NN-SGD	kNN (k=3)	RIPPER	C4.5	RF
TPR (%)	41	91.5	92.8	94.2	94.5	94.5	94.2
	6	90.4	91.8	95.4	95.2	92.6	94.6
FPR (%)	41	12.3	10.7	9.0	8.5	8.6	9.1
	6	15.0	12.8	6.8	7.5	11.2	7.1
ACC (%)	41	91.52	92.77	94.20	94.52	94.51	94.21
	6	90.41	91.78	95.36	95.15	92.61	94.61
PRE (%)	41	91.8	93.1	94.5	94.8	94.8	94.6
	6	91.3	92.4	95.5	95.4	93.0	94.6
ROC (%)	41	93.3	91.0	93.9	93.1	94.6	99.3
	6	97.9	89.5	94.6	93.8	94.0	97.1
RMSE (%)	41	29.02	26.88	23.19	23.41	23.26	19.73
	6	30.91	28.67	20.89	22.08	25.41	20.64

Table 8: Performance of ensembles on "Uni Corr Test" and "Red Uni Corr Test" test datasets using 41 and 6 features

Evaluation Metrics	# Features	Ensemble of Classifier (Vote)				
		Average	Product	Majority Voting	Minimum	Maximum
TPR (%)	41	94.3	93.7	94.3	93.7	91.0
	6	93.5	93.2	93.6	93.2	92.1
FPR (%)	41	8.9	9.5	8.9	9.5	13.7
	6	10.3	11.0	10.1	11.0	12.7
ACC (%)	41	94.31	92.66	94.31	92.66	90.99
	6	93.48	91.12	93.60	91.19	92.06
PRE (%)	41	94.7	93.9	94.7	93.9	91.6
	6	93.9	93.8	94.0	93.8	92.8
ROC (%)	41	99.2	91.7	92.7	91.7	99.0
	6	97.9	90.3	91.8	90.3	98.0
RMSE (%)	41	21.94	25.19	23.84	25.19	21.26
	6	22.74	25.98	25.31	25.98	21.78

(99.87%), PRE (99.9%), ROC (100.0%) and TBM (0.08 sec) but TTM is reduced by 67.76% from 15781.1 sec to 5087.73 sec (Table 3). The classifiers kNN, C4.5 and RF achieved same ROC of 100.0%, the classifier C4.5 achieved same TPR (99.9%), PRE (99.9%) and ROC (100.0%) on original and reduced features set. Performance of NB and NN-SGD is slightly higher for original features set, whereas performance of RIPPER on reduced 6 features

Table 9: Performance of classifiers on "Red Uni Train" training dataset for 6 features using 10-fold cross validation

Evaluation Metrics	Classifiers					
	NB	NN-SGD	RIPPER	C4.5	RF	HEIC
TPR (%)	95.1	97.1	99.8	99.8	99.9	99.8
FPR (%)	6.1	3.7	0.2	0.2	0.1	0.2
ACC (%)	95.12	97.14	99.83	99.83	99.9	99.83
PRE (%)	95.1	97.2	99.8	99.8	99.9	99.8
ROC (%)	99.3	96.7	99.8	99.9	100	99.8
TBM (sec)	0.22	105.87	42.57	3.12	32.68	184.91
RMSE (%)	21.97	16.92	4.05	3.82	2.85	4.09

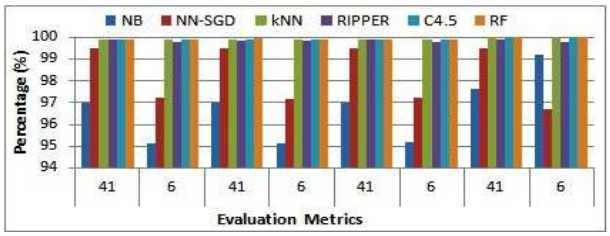


Figure 3: Performance comparison of classifiers in terms of TPR, ACC, PRE and ROC on 41 & 6 features

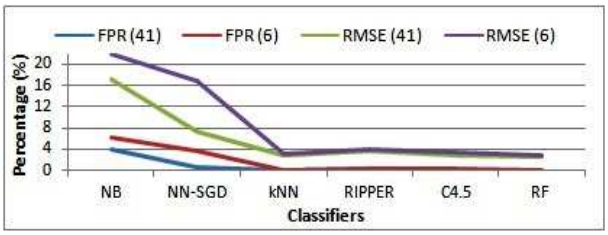


Figure 4: Performance comparison of classifiers in terms of FPR, RMSE on 41 & 6 features

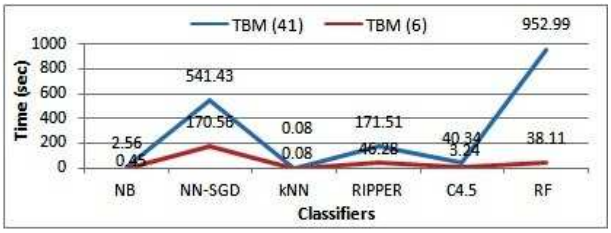


Figure 5: Comparison of TBM for 41 & 6 features in sec

set is near to that of original one. Comparative graph of the performance of classifiers for 41 and 6 features set are shown in terms of TPR, ACC, PRE, and ROC in Figure 3, FPR and RMSE in Figure 4. As can be seen from Table 3, TBM and TTM for original 41 features set are remarkably higher than that of reduced 6 features set. The TBM is reduced by approximately 68-96% except for kNN and TTM is reduced by approximately 40-94% for 6 features set. The Figure 5 and 6 show comparative graph for TBM and TTM on 41 and 6 features respectively. The performances of classifiers were evaluated using test datasets("Uni Test" and "Red Uni Test") are illustrated in Table 5. As can be seen, performances of NB and NN-SGD classifiers in testing phase (Table 5) are little bit higher with comparison to the performance of these classifiers in training phase (Table 3), whereas classifiers kNN, C4.5, RIPPER and RF performed near to equal in training and testing phase. From Figure 3, 4, 5 and 6, it can be observed that selection of optimized features set consume less computation time in training and testing phase and also maintain the same classification performance as of original features set. Therefore, feature selection approach helps to build lightweight NIDS suitable for real time and on-line processing by selecting non-redundant, informative and relevant features.

Among six classifiers, kNN yielded highest TTM (5087.73) which is remarkably very high and will increase the computation time of the ensemble and in turn degrade the performance of HyFSA-HEIC. It is also not suitable for high volume network traffic for real time and on-line processing. Therefore, it was not selected as base classifier in ensemble. As a result, 5 base classifiers—NB, NN-SGD, RIPPER, C4.5 and RF were selected to form the ensemble based on the evaluation. The decision of these 5 classifiers in ensemble is combined by using 5 algebraic combination rules of classifier fusion approach—*Average, Product, Majority Voting, Minimum,* and *Maximum* to produce final decision. Finally, five ensembles were formed. Table 4 illustrates the results of these 5 ensembles on training datasets ("Uni Train" and "Red Uni Train").

Among all 5 ensemble models, ensembles with average and majority voting combiners achieved equally best performance in terms of TPR (100.0%), FPR (0.0%), ACC (99.97%), PRE (100.0%), and ROC (100.0%) on original dataset and TPR (99.9%), FPR (0.1%), and PRE (99.9%) on reduced dataset. Majority voting combiner outperformed in ACC (99.91%), average combiner outperformed in ROC (100.0%) on reduced dataset, majority voting combiner has lowest RMSE among all ensemble models as 1.85% and 3.06% for 6 and 41 features respectively from the results of Table 4. Maximum combiner has lowest performance on all evaluation metrics except

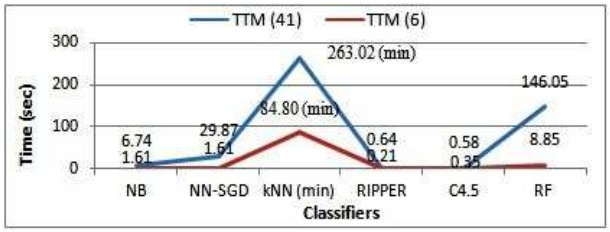


Figure 6: Comparison of TTM for 41 & 6 features (kNN in min)

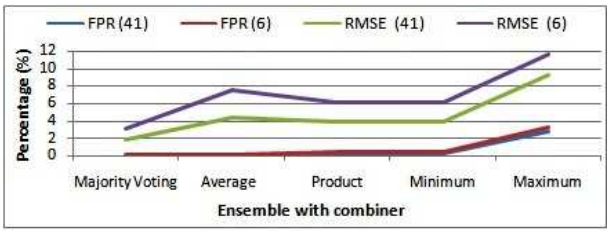


Figure 8: Performance comparison of ensembles in terms of FPR and RMSE on 41 and 6 features set.

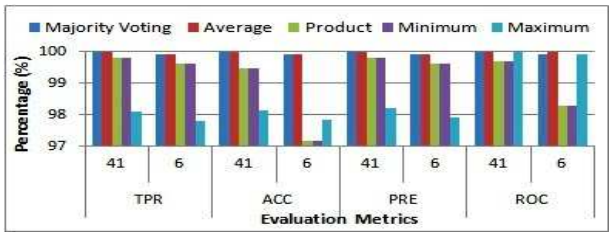


Figure 7: Performance comparison of ensembles in terms of TPR, ACC, PRE and ROC on 41 and 6 features set.

Figure 9: Comparison of Time-span to Build the Model (TBM) for ensembles on 41 and 6 features in seconds.

ROC (100.0%) on 41 features and TBM (253.69 sec) on 6 features. Whereas performances of product and minimum combiners degraded due to unclassified instances of 275 (0.38%) and 1800 (2.47%) on 41 and 6 features respectively but achieved lowest error rate except majority voting combiner (Table 4). Therefore the best performing combining rule for ensemble model is majority voting based on overall performance. TBM and TTM for all combiners of ensemble models are almost the same on 41 as well as on 6 features set. From Table 4, it is observed that TBM and TTM of ensembles are drastically reduced by approximately 58-64% and 42-56% respectively for 6 features set. Figure 7 shows performance comparisons of ensembles in terms of TPR, ACC, PRE, and ROC, Figure 8 for FPR and RMSE, Figure 9 for TBM and Figure 10 for TTM for 41 and 6 features. The performances of ensembles in testing phase on test datasets ("Uni Test" and "Red Uni Test") as illustrated in Table 6 are almost the same to the performances in the training phase (Table 4). The performances of product and minimum combiners degraded due to unclassified instances of 255 (0.35%) and 1745 (2.40%) on 41 and 6 features respectively but achieved lowest error rate except majority voting combiner (Table 6) in the testing phase also.

The experiments were also conducted to measure the performance of classifiers and ensembles on "Uni Train" and "Red Uni Train" training datasets for 41 and 6 features using 10-fold cross validation. It was found that the performance of classifiers and ensembles using 10-fold cross validation were almost same to the performance in the training and testing phase for 41 and 6 features. As can be seen from the results of Tables 3, 4, 5, 6 and 9 of NB, NN-SGD, RIPPER, C4.5, RF and HEIC for 6 features.

The performance of classifiers and ensembles also tested on "Red Uni Corr Test" test dataset for reduced 6 features are near to same the performance on "Uni Corr Test" for original 41 features on all evaluation metrics as illustrated in Tables 7 and 8. Hence the proposed system also achieves near to equal performance on reduced 6 features on this test dataset. The performance of classifiers and ensembles in training phase (Tables 3 & 5) on "Uni Train" and "Red Uni Train" training datasets and testing phase (Tables 4 & 6) on "Uni Test" and "Red Uni Test" for 41 and 6 features are at the higher side than that of testing phase (Tables 7 & 8) on "Uni Corr Test" and "Red Uni Corr Test" test datasets for 41 and 6 features. The reason for this is the test dataset ("Corrected Test") is not from the same probability distribution as the training dataset ("10 % KDD") as it includes additional 17 novel attack types that are not present in training dataset. It makes the system more realistic to perform on real time data.

On the comparison of results of 5 ensemble models (Tables 4 & 6) and 6 classifiers (Tables 3 & 5) based on different evaluation metrics with selected 6 features has shown that ensemble with majority voting combiner outperformed other ensemble models and individual classifiers and thus more reliable and capable for NIDS and hence chosen as ensemble model for HyFSA-HEIC. The performance comparison of individual classifiers NB, NN(SGD), RIPPER, C4.5, RF and HyFSA-HEIC in terms of TPR, ACC, PRE, ROC is shown in Figure 11 and in terms of FPR and RMSE in Figure 12. The results strongly indicate that by employing feature selection approach as preprocessing step and heterogeneous ensemble of intelligent

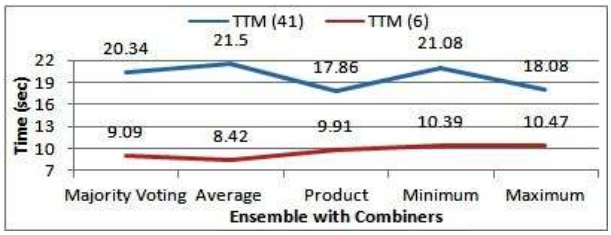


Figure 10: Comparison of Time-span to Test the Model (TBM) for ensembles on 41 and 6 features in seconds.

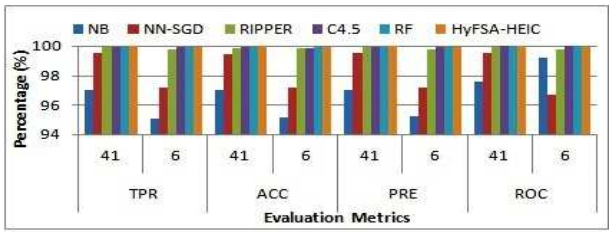


Figure 11: Performance comparison of classifiers NB, NN-SGD, RIPPER, C4.5, RF and HyFSA-HEIC.

classifiers in model building enhance the performance of HyFSA-HEIC. It has been enhanced in terms of TRP (99.9%), ACC (99.91%), PRE (99.9%), ROC (99.9%), with extremely low FPR (0.1%) & RMSE (3.06%) and has faster building and testing time than the ensemble with full features set.

7 Conclusion and Future Work

The aim of this work is to propose Hybrid Feature Selection Approach – Heterogeneous Ensemble of Intelligent Classifiers (HyFSA-HEIC) for network intrusion detection and to demonstrate that this system can enhance the accuracy and efficiency of the system as well as reduce the false positive rate, error rate, training and testing time. It hierarchically integrates hybrid feature selection approach (HyFSA) with heterogeneous ensemble of intelligent classifiers (HEIC). The main challenging issues arise in IDS are to handle large-scale high dimensional dataset and maximizing overall accuracy and less false alarm. The HyFSA-HEIC addresses these issues by incorporating hybrid feature selection approach (HyFSA) and heterogeneous ensemble of intelligent classifiers (HEIC). The heterogeneous ensemble built in this work employed five diverse accurate intelligent classifiers—NB, NN (SGD), RIPPER, C4.5 and RF and their decisions were combined by utilizing majority voting of elementary combiner based on algebraic combination rule. This ensemble was built on using only 6 selected features i.e. only 15% of original 41 features. Several experiments were performed to compare the HyFSA-HEIC with other ensembles and individual classifiers with and without applying feature selection approach. "KDD Cup 1999" and "Corrected Test"

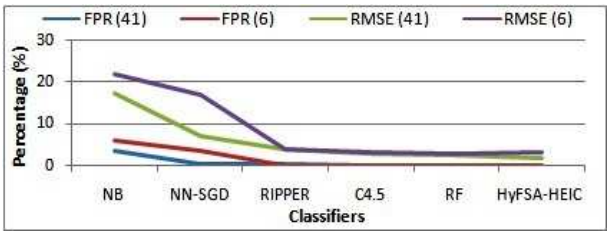


Figure 12: Performance comparison of NB, NN-SGD, RIPPER, C4.5, RF and HyFSA-HEIC.

datasets have been utilized to train, and test the methods and HyFSA-HEIC used in this work. The results show that HyFSA-HEIC outperforms other methods with true positive rate (99.9%), accuracy (99.91%), precision (99.9%), receiver operating characteristics (99.9%), and low false positive rate (0.1%) and root mean square error rate (3.06%) with minimum number of selected 6 features. It also reduces the training time by 50.79% and testing time by 55.30% on reduced features set. The classifiers used in proposed HEIC are applicable to both numerical and categorical features as well as on large dataset, which is practically advantageous for real time intrusion detection. In conclusion, integrating feature selection approach to the heterogeneous ensemble of intelligent classifier improve the detection rate, accuracy, precision, and receiver operating characteristics and reduce the false alarms and error rates with minimum computation time.

Due to continuous increase of intrusion or attack and ever-growing network traffic in computer networks, there has been an endless requirement for improvement in the performance of NIDS especially in terms of low false rate and minimum computation time. Therefore, we anticipate enhancements in the performance of the proposed heterogeneous ensemble method in terms of high accuracy, low false rate, low error rate and minimum computation time. This can be achieved by inducing higher diversity among the classifiers in ensemble as well as by investigating other classifiers for ensemble or ensemble methods. The proposed system is capable of classifying between attack and normal traffic connection, but lack in dealing with further classification of specific attack type. The extension of this work is to further classification of attack into four classes—DoS, Probe, U2R and R2L i.e. multi-class classification and determination of optimal features set for each attack type for network intrusion detection.

References

[1] B. Agarwal and N. Mittal, "Hybrid approach for detection of anomaly network traffic using data mining techniques," in *Proceedings of 2nd International Conference on Communication, Computing and Security, Procedia Technology*, vol. 6, pp. 996–1003, 2012.

[2] Amrita and P. Ahmed, "A study of feature selection methods in intrusion detection system: A survey," *International Journal of Computer Science Engineering and Information Technology Research*, vol. 2, no. 3, pp. 1–25, 2012.

[3] Amrita and P. Ahmed, "A hybrid-based feature selection approach for IDS," in *Proceedings of 5th International Conference on Networks and Communications (NETCOM'13)*, vol. 284, pp. 195–211, Chennai, India, Dec. 2013.

[4] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[5] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'10)*, pp. 177–187, Paris France, Aug. 2010.

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

[7] R. Chitrakar and H. Chuanhe, "Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and Naïve Bayes classification," in *Proceedings of 8th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'12)*, pp. 1–5, Sep. 2012.

[8] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the 12th International Conference on Machine Learning*, pp. 115–123, Paris France, 1995.

[9] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Proceedings of the Third International Conference on Machine Learning*, pp. 74–81, Dalian, China, June 2001.

[10] T. G. Dietterich, "Ensemble methods in machine learning," *Multiple Classifier Systems, Lecture Notes in Computer Science*, vol. 1857, pp. 1–15, 2000.

[11] I. El-Henawy, H. M. El Bakry, H. M. El Hadad, "A new muzzle classification model using decision tree classifier," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 12–24, 2017.

[12] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, pp. 1022–1029, Paris France, 1993.

[13] T. F. Ghanem, W. S. Elkilani, and H. M. Abdul-Kader, "A hybrid approach for efficient anomaly detection using meta heuristic methods," *Journal of Advanced Research*, vol. 6, no. 4, pp. 609–619, 2014.

[14] V. Golmah, "An efficient hybrid intrusion detection system based on C5.0 and SVM," *International Journal of Database Theory and Applications*, vol. 7, no. 2, pp. 59–70, 2014.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, 2009. (`http://www.cs.waikato.ac.nz/ml/weka/`)

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

[17] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[18] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[19] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[20] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

[21] S. W. Lin, K. C. Ying, C. Y. Lee, and Z. J. Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," *Applied Soft Computing*, vol. 12, no. 10, pp. 3285–3290, 2012.

[22] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Boston: Kluwer Academic, 1998.

[23] M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," in *Proceedings of the International Conference on Communication Technology and System*, vol. 30, pp. 1–9, 2012.

[24] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[25] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[26] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Systems with Applications*, vol. 39, no. 1, pp. 129–141, 2012.

[27] A. Tayal, N. Mishra and S. Sharma, "Active monitoring & postmortem forensic analysis of network threats: A survey," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 49–59, 2017.

[28] S. Thirumuruganathan, *A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm*, World Press, 2010.

[29] UCI KDD, *KDD Cup 1999 Intrusion Detection Dataset*, Oct. 28, 1999. (`http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`)

[30] UNB, *NSL-KDD Network-based Intrusion Detection Dataset*, Apr. 29, 2017. (`http://nsl.cs.unb.ca/KDD/NSL-KDD.html`)

[31] H. Zhang, "The optimality of Naïve Bayes," in *Proceedings of The Seventeenth International Florida Artificial Intelligence Research Society Conference*, pp. 17–19, Miami Beach, 2004.

Biography

Amrita is an Assistant Professor in Department of Computer Science and Engineering at Sharda University, Greater Noida, INDIA. She received her M.Tech. in Computer Science from Banasthali Vidyapith, Rajasthan. She is currently pursuing her Ph.D. in Computer Science and Engineering from Sharda University, Greater Noida (U.P.). She has more than 15 years of experience in Academics, Software Development Industry and Government Organization.

Kiran Kumar Ravulakollu is working with Sharda University of INDIA as Assistant Professor with interests in sensory networks, robotics, biologically inspired multimodal behaviour modelling, ambient intelligence, and sensory network design along with visual and auditory information processing. He has more than 5 years of research experience in Hybrid Intelligent Systems area at Centre for Hybrid Intelligent Systems, University of Sunderland, UK. He has received his Ph.D. degree from University of Sunderland, UK.