AMSE Association for the Advancement of Modelling & Simulation Techniques in Enterprises
IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Advances in Modelling and Analysis B
Vol. 63, No. 1-4, December, 2020, pp. 10-19
Journal homepage: http://iieta.org/journals/ama_b

Hybrid Architecture for Distributed Intrusion Detection System Using Semi-supervised Classifiers in Ensemble Approach

Shraddha R. Khonde[1,2*], Venugopal Ulagamuthalvi[2]

[1] Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai 600119, India
[2] Department of Computer Science and Engineering, M.E.S. College of Engineering, Pune, S.P. Pune University, Pune, 411001, India

Corresponding Author Email: khondeshraddha21@gmail.com

https://doi.org/10.18280/ama_b.631-403

Received: 26 February 2019
Accepted: 11 November 2020

Keywords:
intrusion detection system, Gini index, feature selection information security, ensemble, network security, semi-supervised, machine learning

ABSTRACT

Security of data is becoming a big treat today because of modern attacks. All the data passing through network is at risk as intruders can easily access and modify data. Security to the network is provided using Intrusion Detection System (IDS) which helps to monitor and analyze each packet entering or passing through the network. In this paper hybrid architecture for IDS is proposed which can work as an intelligent system in distributed environment. Proposed system makes use of semi-supervised machine learning classifiers into an ensemble approach. Classifiers used are Support vector machine, decision tree and k-nearest neighbor. Ensemble of this classifier is done and final prediction is given by majority voting algorithm. This system makes use of feature selection technique to reduce number of features used for training various classifiers. Experiments are conducted on NSL-KDD dataset. From results it is observed that ensemble technique increases accuracy by 3% and reduces false alarm rate by 0.05. System performance improves if used in ensemble approach as compare to individual classifier.

1. INTRODUCTION

Security to the network is provided by Firewall which helps in securing network by preventing entry of unreliable packets in the network. Intrusion detection system mostly provides a supplementary protection to the firewall as is not the alternative for it [1]. Now a days as most of the intruders are using modern attacks most of the unreliable packets gets undetected and passed through firewall. To avoid this researcher observes a need of smart IDS which will detect all the packets which are detected as malicious activity. Now day's network security can be provided with various emerging areas such as artificial intelligence, internet of things, data mining, cloud computing and machine learning. Most of the IDS make use of various frameworks to identify various modern attacks [2]. Various machine learning algorithms as supervised, semi-supervised and unsupervised algorithms are used to improve performance of IDS and convert it into an intelligent system as compared to firewall. These techniques help to improve detection rate and accuracy of classifiers and apparently performance of IDS. With the aim of creating an intelligent DIS which will resolve issues of security regarding web sites, personal computers and networks. To build an intelligent IDS one should make use of various algorithms from data mining, artificial intelligence and machine learning. Benefit of using these algorithms is to recognize different novel attacks entering in the network with good detection rate and reduce misclassification [3]. Most of the traditional IDS misclassify attacks as it goes unidentified this issue can be resolved by intelligent IDS. IDS use two different types of attack detection methods as elaborated below.

1.1 Pattern-based IDS

Signature based IDS (pattern based) is a detection method which make use of signature database. Signature database consist of signatures of various modern attacks, which are the definitions of attacks stored in same structure. Signature can be found and stored in any format and mostly known as foot prints obtained after attack happen on the network. Each attack once happen in network leaves its foot prints behind which are used to create a signature of that attack in specific format. Examples of foot prints are basically the sequence of actions followed by other actions on a node, change in payload, longer response time or request from same host etc. This method mostly works on matching of signatures stored in dataset. If matching of signature is found then the packet is considered as attack or malicious activity. Otherwise if matching to signature from dataset is not found packet is considered as normal packet and passed in the network. Signature should be in database when this method is used by IDS for attack detection. Encounter for IDS following this method of detection is to collect signature of modern attacks. If signature of modern or novel attacks is not available IDS performance decreases. Pattern based IDS oversight most of the not known and fresh attacks. This is due to the lack of signatures for these attacks in database. Enhanced accuracy is observed in detection of known attacks as compared to unidentified attacks [4]. Most of the researchers use various classifiers along with signature based detection to increase accuracy of IDS. In some networks where all the attacks are known and no novel attacks are happening in the network signature based detection is used. To

use signature based detection with good performance modern datasets such as UNSW NB15, CICIDS 2017 can be used.

1.2 Behavior-based IDS

To overcome constraint of pattern based IDS and increase detection rate of the IDS to work intelligently most of the researchers use anomaly or behavior based detection. In case of missing foot records analysis of packet becomes tedious. Solution for it is to use behavior detection based IDS. Anomaly based IDS checks for abnormal behavior of the packets entering in the system. Normal behavior can be considered as number of packets entering in the network, size of the packets, source and destinations mentioned in the packets, source and destination ports used, payload field values and many more. Packet entered in the network is analyzed according to rules set for normal behavior of any packet. It entered one equals to rules consider as normal else abnormal and declare as attack. On revealing of attack generation of alarm process gets initiated. Limitation of this type of detection is slight deviation from the normal behavior tends towards alarm generation. In this type of method wrong alarm generation is more as compared to signature based method. False alarm intensification liable towards deprived detection rate and system performance [4]. Anomaly detection helps in finding novel attacks provided exact behavior of the network packets are identified. In most of the cases researchers find that it is very complex task to find exact behavior of any network packets. This is the main reason why anomaly detection method has more misclassification which tends towards increasing false alarm rate of IDS.

Organization of paper is, section 2 elaborated about the literature survey for proposed work. Section 3 explains proposed system methodology followed by experimental analysis with detail explanation of dataset used in section 4. Section 5 focuses on results and performance evaluation of the proposed system with various parameters. Section 6 gives final remarks as conclusion and future scope for further research in same area.

2. RELATED WORK

Identification of various modern attacks as to secure network from various malicious activities most of the researchers make use of various emerging techniques. Most of the researchers use several machine learning algorithms, data mining, random forest, support vector machines, cloud computing, neural network [5-7], clustering methods with artificial algorithms [8], Genetic algorithms [9], and ensemble approach. It is used to expand performance of detection and minimize false alarm rate. IDS performance can be degraded as size of dataset increases due to various upcoming modern attacks [10]. Detection rate of IDS also depends on the dataset used for training classifiers used. The KDD-99 dataset is elaborated in depth [11]. KDD dataset includes 4 types of attacks. Host, basic are two types along with content and last one is traffic. Recognition rate and rate for misclassification is used for evaluation. Authors create four clusters, which are used to create 15 subsets. To enhance enactment class dominance is used. Feature selection based support vector machine with combination of k-means and information is used [12] for improving IDS performance. Feature importance is calculated using information gain which used to reduce features to 23 for improving performance.

Authors use random forest and j48 for finding performance of system [13]. Model was implemented using random forest classifier. For training NSL-KDD standard dataset was used. Clustering of dataset is done using various types of attacks. Preprocessing is done using symmetric uncertainty measure. This is done for reducing number of features used. Performance for detection rate and false rate is analyzed by creating 100 trees. Various feature selection methods as gain ratio; correlation and information gain is used to evaluate performance on 41 features of KDD [14]. Decision tree classifier is used along with attribute ratio to compare performance with information gain. Same process is followed by authors in gain ratio as well. Results verified that accuracy improves when reduces 22 features used for KDD. Ensemble of SVMs is also an efficient approach used by many researchers for performance improvement [15].

Combination for ensemble is used with KNN, decision tree and artificial neural network [16]. To train and test various experiments DARPA dataset along with mentioned classifiers are used. Comparison of all classifiers is presented to check performance in terms of recognition rate and misclassification. Another ensemble of SVM and KNN is used along with KDD 99 dataset is explained by Amin and Reaz [17]. Practical swarm optimization and weighted majority algorithm is used for experiment analysis. More explanation on weighted majority algorithm is given by Littlestone and Warmuth [18] which use majority voting for final prediction. A modular ensembling technique classifying available service like log files, audit, mail service, web service etc. into a separate category is given by Giorgio et al. [19].

Authors focuses on online and offline mode of IDS framework based on modeling of random forest classifier [20]. While modeling this classifier real time traffic was used to create dataset. This created dataset is used to train random forest classifier. Authors check performance of classifier in online and offline mode. Results proved that this classifier takes more computation cost in online mode as compared to offline mode when tested on real network traffic. Trees in classifier are deploying using bootstrap samples from preprocessed dataset.

Another unsupervised approach with the help of data mining in elaborated by Jungsuk et al. [21]. This architecture works in different phases. Deployment starts with filtration phase followed by clustering and then modeling. Main motto behind this framework is to take input from user for training and testing. Filtration process mainly focused on attack packets which will eliminate these packets from dataset. Algorithm used for elimination is notion of density which based on formula where ratio of attack packets to the network packets was checked. Researchers also use an approach tending towards various clustering algorithms. Horng et al. [22] make use of BIRCH algorithm based on clustering. Based on the principal types available in KDD'99 dataset the algorithm divides it into number of parts. Each part represents one of the types of attack available as probe or dos. It also covers u2r and r2l in parts. Normal traffic works as a separate part in to the BIRCH process where each part is nothing but cluster used by algorithm. BIRCH deploys various clustering trees based on each part created in the process.

Most of the researchers are now days creating own customized dataset. Sources can be a heterogeneous in manner used for data collection [23]. Most of the sources can be log files, audit files, and packets entering in the network from

various senders. All this data is extracted and used in creation of new dataset and labeled by the expert from the domain. Dataset can be further divided in training and testing part which can be also used for validation using semiautomatic methods. Authors make use of k means to eliminate extra features from the dataset. Validation can be done for checking system performance which makes use of ensemble of decision tree and other classifiers. Weighted mean algorithm is used for partitioning and validating each part of dataset.

From related work of many researchers it is observed that most of the datasets are not having signatures of novel and modern attacks. This leads towards poor performance of intrusion detection system. If the signatures of modern attacks are not available in the dataset then signature based detection approach shows poor performance. Anomaly detection approach also leads towards more misclassification and turns into generation of more false alarm. To overcome limitation of both the detection methods proposed system works in hybrid fashion where both detection methods are used. It is also observed that most of the classifiers show biased prediction when used individually in network as IDS algorithms. To provide solution to it proposed system makes use of ensemble approach where number of classifiers is working together to find final prediction about packet as normal or attack. Proposed system makes use of classifiers as decision tree (DT), support vector machine (SVM) and k-nearest neighbor (KNN). Final prediction is generated using majority voting algorithm. Ensemble approach provides improved performance of IDS over individual classifiers. Table 1 shows summary of related work.

Table 1. Related work in intrusion detection system

Reference Number	Methodology	Classifiers
[6]	Supervised, Ensemble	SVM, RF
[12]	Supervised, Ensemble	SVM, KNN
[13]	Supervised, Ensemble	RF, j48
[14]	Supervised,	DT
[15]	Supervised, Ensemble	Multiple SVM
[16]	Supervised, Ensemble	ANN, SVM, DT, KNN
[17]	Ensemble	SVM, weighted majority, KNN
[19]	Ensemble	K means and SVM
[20]	Supervised, Ensemble	RF
[21]	Supervised, Ensemble	Clustering, SVM
[22]	Supervised, Ensemble	BIRCH, SVM
[23]	Semi-Supervised	DT, weighted mean and k means clustering

3. PROPOSED METHODOLOGY

Proposed system architecture is a distributed IDS working in distributed environment. NSL KDD dataset is used for training the classifier and testing all classifiers used while implementing system proposed in this paper. A benchmark dataset used in this dataset is NSL KDD. This dataset consists of total 41 features and four different types of attacks are covered in this. Signatures of these four types of attacks are stored in this dataset along with label mentioning type of attack or normal packet signature. This dataset consists of more than lakh records and some are redundant in nature. To remove redundant data and clean dataset, preprocessing is done. Preprocessing phase is added in architecture to remove invalid instances present in the dataset. As the proposed system is for distributed environment only features required for attack detection is used. Feature selection decreases the time require for training and analyzing any packet by classifier. Gini Index feature section technique is used in proposed system to reduce number of features used to train classifiers. Gini Index is used to calculate impurity of features. After calculating Gini Index of all features, the one whose impurity is less is considered in reduced set.

The features having more impurity are removed from the final feature set used for training classifiers. Training of classifier is trained using reduced features. Total three classifiers as decision tree, support vector machine and k-nearest neighbor. All these classifiers are trained and testing for attack detection. The prediction of this classifier is passed to the ensemble approach where an algorithm is used which use concept of voting based on majority. This algorithm is responsible to declare final output predicted. Ensemble approach is used in proposed system to avoid biased output of individual classifier. To improve system performance ensemble method is used. This will improve performance of IDS as a system as well as performance of individual classifier. Final prediction given by majority voting algorithm is that the packet entered in system is malicious or normal. For example if two out of three classifiers have voted as malicious packet and one as normal packet then according to maximum votes packets are finally predicted as malicious. Proposed methodology is shown in Figure 1.

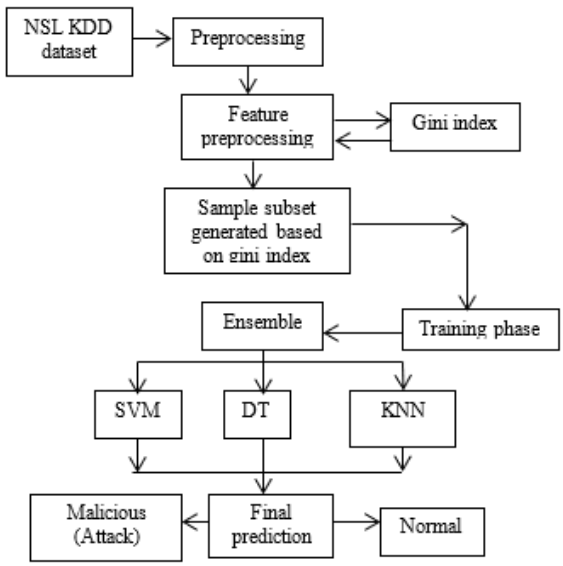


Figure 1. Proposed methodology

Next step followed in proposed system is after detection of attacks. Here administrator is responsible to imitate the process. Once malicious activity is detected by classifiers an alert is generated for administrator. Administrator need to take action based on level of activity. He can block certain senders in the network or can broadcast the address to each node connected in the network such that destination nodes can avoid accepting data from such senders. Administrator make use of dashboard to check status of each performance parameter such as misclassified packets, correct classified, rate of detection, and accuracy. Proposed system works in real time environment so KDD extractor is used to capture and extract packets so that testing can be performed. These files are passed for preprocessing to generate a reduced subset of features and then passed for testing using ensemble classifiers.

3.1 Selection of important features

One of the important steps in training of classifiers is to choose right features for detection such that improvement can be recognized. Most of the standard datasets available for intrusion detection are consist of invalid instances as well as noise. This dataset is passed for preprocessing before sending it to feature selection. In proposed system Gini Index feature selection technique is used. This technique helps in finding impurity of each feature. Impurity is nothing but the importance of the feature in identifying attacks. The feature having more impurity is less important as it does not consist of valid values to form signature and can be of less use in attack detection. Time taken by classifier for training is very important in distributed environment which is not possible to reduce if all features were used. To reduce this processing time features need to minimize which can be done with the help of various feature selection techniques. Anyone can reduce features up to √A, where A can be number of features available in any benchmark dataset. According to observations from various surveys, reduce features helps in reducing processing time as well as training time of classifier. It also helps to enhance accuracy and detection rate [24, 25]. The main reasons to use feature selection are:

- Quick training of classifier, reduce over fitting
- Less number of feature handling so less complexity
- Accuracy increases as proper subset is selected.

Most of the researchers use different feature choice techniques such as principal component analysis, correlation analysis, linear discriminant analysis, chi-square, information gain, Gini index and many more for dimensionality reduction. Impurity of feature can be calculated using Gini Index. It is basically used to find partition subset of features. Final impurity of any feature can be calculated by summing up classifiers features selection. Every feature associate with it an importance value called as feature importance value (*FIV*). Eq. (1) shows the formula used for calculating Gini index. Gini index is used by many researchers to compute importance of each feature available in NSL-KDD dataset. Importance of all features are calculated and used to find reduced subset sample.

$$gini\ index\ (f_i) = 1 - \sum_{i-1}^{m} P_i^2 \qquad (1)$$

where, f_i is the Gini index of i^{th} feature
m is KDD dataset features that is 41 in total
P_i is the feature probability

In architecture, preprocessing phase used to calculate importance value of each feature (*FIV*). It is calculated by summing up all classifiers Gini index as per in Eq. (2).

$$FIV = 1 - \sum_{i-1}^{n} gini\ index\ (f_i) \qquad (2)$$

where, *Gini index (fi)* is *fi* Gini index
n is feature in total

Features having less impurity and high feature importance value are used to create subset for training classifiers. Table 2 shows feature importance value (*FIV*) for each feature of NSL KDD dataset.

Table 2. *FIV* for NSL KDD features

Feature Number	*FIV* score	Feature Number	*FIV* score	Feature Number	*FIV* score
1	0.67	15	0.31	29	0.56
2	1.01	16	0.56	30	0.89
3	1.3	17	0.41	31	0.53
4	0.78	18	0.56	32	1.26
5	1.6	19	0.24	33	0.52
6	0.76	20	0.32	34	0.20
7	0.42	21	0.44	35	0.28
8	1.9	22	0.82	36	1.12
9	0.45	23	1.15	37	0.88
10	1.25	24	0.46	38	0.56
11	0.56	25	0.89	39	0.88
12	0.78	26	0.45	40	0.44
13	0.98	27	0.69	41	0.40
14	0.79	28	0.45		

According to *FII* some features having highest importance are selected for reduced dataset for training classifiers. Out of all features of NSL KDD, 21 features are used to train classifiers.

3.2 Ensemble classifiers

In proposed system are support vector machine, k-nearest neighbor and decision tree are used in ensemble. Ensemble approach provides improvement in prediction of classifiers as compared to individual classifiers. Figure 2 shows graphical representation of ensemble approach.

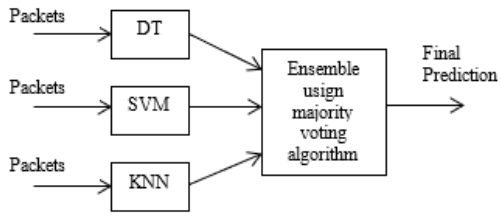


Figure 2. Ensemble approach

Algorithm called majority voting helps in predicting final output as attack or normal. This algorithm gathers votes from every individual classifier and according to majority final prediction is obtained. Detailed explanation of each classifier is explained in next section.

3.2.1 DT - decision tree

Tool used to support decisions made by users is called a decision tree algorithm. This algorithm is from the area of data mining which helps in mining data in an efficient format. This algorithm makes use of tree like structure to view users decision in graphical format. Utility of each node is to rectify all the consequences possible upon the choice of specific data. It also shows the outcome of the specific choice and computation or resource cost required. Area of operation research most make of this algorithm as tool where define strategy is very important in decision making so that goal can be achieved. Representation of this algorithm is like a tree or flowchart where each node presents an answer for test taken on attribute. Depend on that the branches are created and nodes are attached one after other. Last node will be the leaf node basically shows the final output or final decision of the user. These nodes are mostly used to represent class labels. Lables are associated with leaf nodes after completing tests on

attributes. The path from root node up to leaf node is used for formation of rules, which can be used in creation of dataset. While doing analysis of decision take the tree like representation of this algorithm helps analyst to get the exact decision. Learning phase of this tree can be done with the help of part of any dataset. The source can be splatted into number of parts and then the tree can be trained. Parts or subsets can be formed according to attributes values or range. Process if followed repeatedly to get a better split for the dataset. Approach used in this algorithm is top-down and finds best split for decision making. Algorithm makes use of various nodes as root node, intermediate node and leaf node. Root nodes are attributes from the dataset. The output of test taken on attribute is shown with the help of intermediate node. Final class labeled is represented using leaf node. Figure 3 shows graphical representation of decision tree.

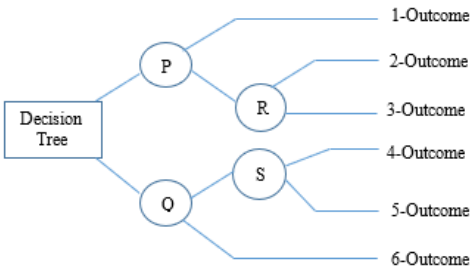


Figure 3. Graphical representation of decision tree

3.2.2 SVM - Support vector machine

This classifier works on the concept of hyper planes. In training 'p' number of hyper planes is created for 'q' number of classification categories. SVM works with classification as well as regression. Classification is based on hyper planes which divides plane into two classes such that instances can be divided into multiple categories. In proposed system five categories are considered as four types of attack signatures are available in NSL-KDD dataset. SVM does not support categorical data. To train SVM using NSL-KDD dataset string type of data need to be converted into numerical data. While testing real time data captured packet is converted according to SVM. As the values of features in NSL-KDD various in between different ranges, data normalization plays important role. If data is not normalized, classifier can provide biased prediction. This leads towards decrease in accuracy. Eq. (3) is used for data normalization in SVM.

$$N_2 = (N_1 \times Minimum)/(Maximum - Minimum) \qquad (3)$$

where N_2 - is the new value

N_1 - is the old value

Figure 4 shows SVM representation using Hyper planes.

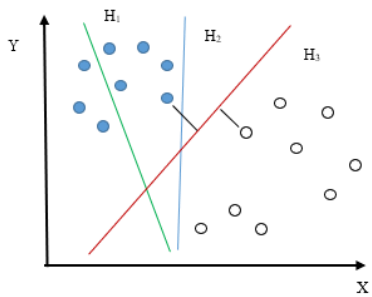


Figure 4. SVM representation using Hyperplanes

3.2.3 KNN – K nearest neighbor

One of the best algorithm in pattern finding is k nearest neighbor. In proposed system this algorithm is used to find specific patterns required for data detection. Method used by this algorithm is to learn with analogy. Entered data is matched with various patterns to find the exact identification of malicious activities. Figure 5 shows graphical representation of KNN.

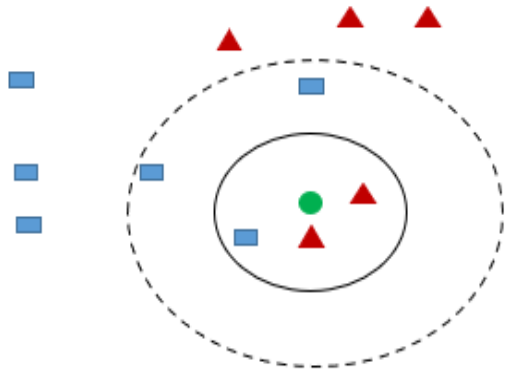


Figure 5. Graphical representation of KNN

This algorithm makes use of n dimensional spaces where n is 41 in proposed system as NSL-KDD dataset has 41 features which are used for attack identification. This n dimensional space represents tuples for the classifiers. Training tuples make use of space of n dimension to store data. For the new packets to test in the system the closet neighbor is found to classify it with the exact matching pattern from n dimension space. Patterns matching to the given input is found and classification takes place. If match odes not found then close neighbor are found and used for classification. Each dimension consist of space has some threshold value which is checked every time new packet entered in the system. If matching goes beyond threshold value it is considered as attack otherwise it is considered as normal packets.

3.3 Hybrid architecture of proposed system

Next step to follow after training all three classifiers using NSL-KDD reduced dataset is the deployment of all classifiers in the distributed environment. To solve the purpose a new distributed system need to be deployed in distributed environment. Figure 6 shows the architecture of the system.

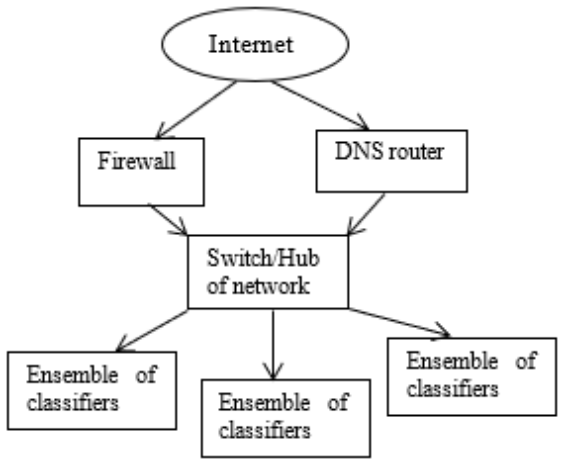


Figure 6. Hybrid model of proposed system

As shown in Figure 6, the internet is used as a first model which is connected to firewall and DNS router. This allows entering packets in the network. First packets were analyzed using firewall. If firewall not able to find any malicious activity it is passed to the switch to transfer to destination. This proposed system firewall works as first line of defense where easily recognizable attacks were identified. The packets which remain unidentified are passed to IDS. In IDS first packets are handled by the central controller that is switch or hub work as a watchman of network. The basic functionality is to extract the features. Feature extraction from the packet is done using KDD extractor. The packets are then further passed to the ensemble classifier where node makes use of all three classifiers to analyze the packet. Final prediction is given by majority voting algorithm which makes predicted input from all individual classifier and according to majority final decision is taken. Ensemble approach increases the accuracy and detection rate of IDS in comparison of individual classifier. Central controller is used to pass the data in the network and packet extraction according to reduced features of NSL-KDD dataset. Less number of features helps in minimum time require to train classifier as well as to reduce processing time in distributed network.

4. EXPERIMENTAL EXPLORATION

4.1 KDD dataset

The KDD dataset is one of the benchmark dataset used in the intrusion detection. This dataset includes three parts as whole KDD which consist of all records of the dataset which consist of near about more than lakh records in total. Second part is 10%KDD which consist of small instances from the whole dataset and mostly used for training of the classifiers. Last part is corrected data which is the latest version of the dataset. Mostly basic KDD dataset consist of redundant and invalid instances which are removed from the new corrected dataset. This dataset has 41 features and more than 1.25 lakhs instances of four different type of attacks along with normal packets are available. All the feature values are stored in symbolic as well as continuous form with the significant range specific in the dataset. Main four types of attacks signatures are available in this dataset as shown in Table 3. After cleaning dataset by removing number of redundant and invalid data dataset is called as NSL-KDD.

Table 3. Attack classification in NSL-KDD dataset

Classification	Attacks
Probe	Mscan, Saint, Port-sweep
Denial of Service (DoS)	Apache2, Neptune, Mailbomb, UDPstorm
User to Root (U2R)	Httptunnel, SQLattack, PS
Romote to Local (R2L)	Named, Sendmail, SNMPGuess, Xlock

4.2 Performance evaluation of classifiers

Performance of the classifier is evaluated with the help of various parameters as false positive rate, accuracy and false negative rate. The calculation and evaluation of all these parameters is depend on the concept of Confusion Matrix. To exactly know how the classifier is classifying data this tool is used. Positive tuples will be created if correct classified by classifier otherwise generation of negative tuples take place. Confusion matrix use following terms,

True Positives (TP) – Classifier classified positive tuples correctly.
True Negatives (TN) – Classifier classified negative tuples correctly.
False Positives (FP) – Classifier classified negative tuples incorrectly
False Negatives (FN) – Classifier classified positive tuples incorrectly.

Table 4 shows the ideal confusion matrix used for performance evaluation.

Table 4. Confusion matrix

		Class predicted by classifier	
Class in actual of classifier		C_1	$\neg C_1$
	C_1	TP	FN
	$\neg C_1$	FP	TN

Confusion matrix uses above mentioned terms for considering classifier performance. This tool is basically used to decide the performance of every classifier considering this correctly and incorrectly classified tuples.

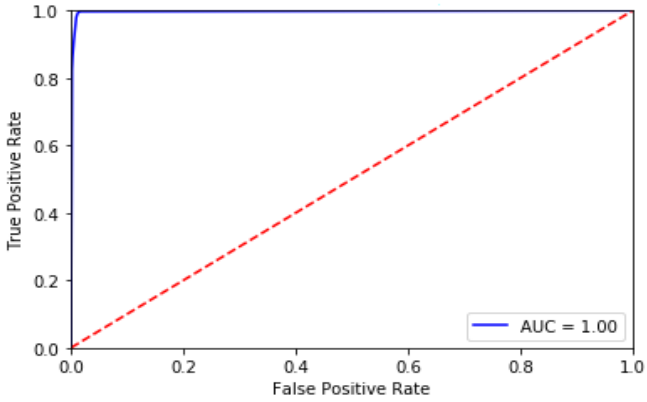
5. RESULTS AND DISCUSSION

As per observations from Table 4 we can say that TP and TN justify correct classification by classifier. While FP and FN shows misclassification done by classifier. Proposed system is tested on real time environment according to confusion matrix. Packets are extracted in converted into .pcap file so that detection of attacks can be possible in less amount of time. To check performance of proposed hybrid system evaluation is done on individual and ensemble approach. Table 5 mentioned results obtained on a single and hybrid classifier tested during real time environment and trained using NSL-KDD reduced dataset.

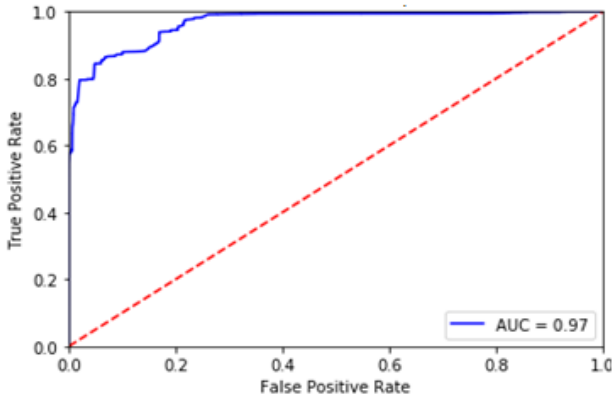
As per the observations represented in Table 5 the values are tested in the real time environment by all classifiers. All the classifiers are trained using NSL-KDD dataset which consist of more than 1.25 lakhs packets. These all classifiers are used to test the incoming network packet. Preprocessing and normalization is performed on the dataset before training classifiers. All classifiers are tested in real time environment on individual basis. This will help in checking performance of individual as well as hybrid classifier. From results presented in Table 5, it can be summarized that hybrid model of proposed system provides more correct prediction as compared to individual classifiers. Confusion matrix is used to find the accuracy and misclassification rate of single as well as hybrid classifier. Graphical representation of results obtained are shown by receiver operating characteristics curves (ROC) in Figure 7.

Table 5. Results obtained for all classifiers according to confusion matrix

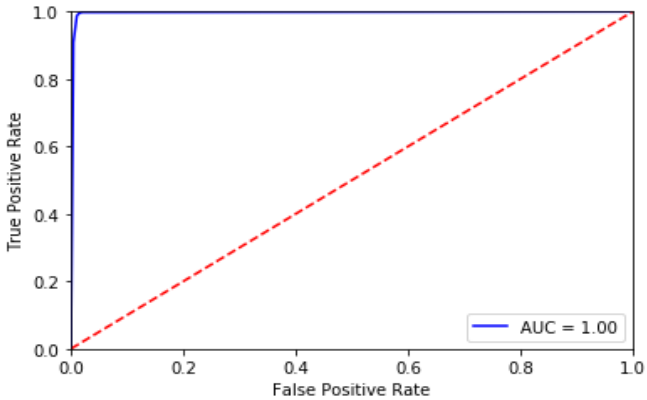
	DT		SVM		KNN		Hybrid	
	C_1	$\neg C_1$	C_1	$\neg C_1$	C_1	$\neg C_1$	C_1	$\neg C_1$
C_1	82954	1106	83577	4058	87105	1196	84189	1240
$\neg C_1$	451	49751	10527	40468	589	47586	221	48921



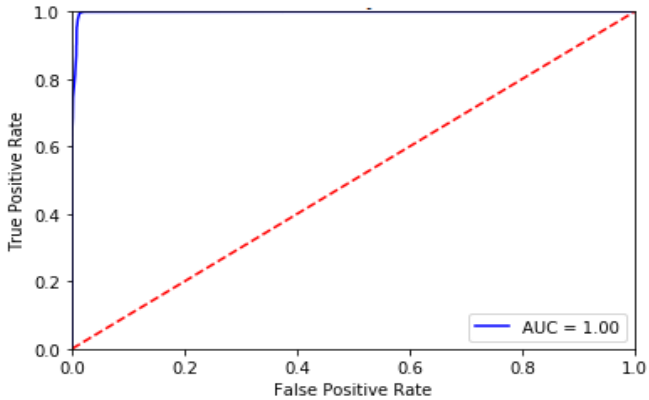
a. DT ROC curve



b. SVM ROC curve



c. KNN ROC curve



d. Hybrid model ROC curve

Figure 7. Receiver operating characteristics curves for all classifiers

Compression of two or more classifiers is done using this tool it is also used to visualise this comparison in the form of curves. It is a useful tool to visualize performance of classifier according to correct and misclassification of packets. Trade-off between false positive rate (FPR) and true positive rate (TPR) is presented using ROC curves. Correctly classified positive tuples are known as TPR and represented as C_1. On the other hand tuples which are misclassified is called ad FPR and represented as $\neg C_1$. These are the misclassified tuples and also known as negative tuples. It is observed from Figure 7(a) that curve is much nearer to the border on left hand side. It is also closed to the top border which means that DT provides good accuracy. Figure 7(b) indicates the curve for SVM, which is farther form the border of left hand side which tends towards less accuracy. ROC curve shown in Figure 7(c) is for KNN classifier. This classifier also shows more accuracy as curves are closer to border. Curve for hybrid model is shown in Figure 7(d). This curve covers maximum area as it is very close to both the borders that is left and top border. By observation we can conclude that hybrid model provides more accuracy as compared to single classifiers. ROC curves make use of confusion matrix to evaluated accuracy provided by each classifier. ROC curve helps to choose best fitted classifier which can be used for implementation of IDS. The one with less false positive rate can be used for implementation of IDS.

Performance evaluation of single and hybrid classifiers is checked using IDS performance evaluators as accuracy, rate of true positive tuples (TPR) and rate of false positive tuples (FPR). Eqns. (4-6) is used for calculating performance parameters according to confusion matrix values.

$$Accuracy = \frac{(TP+TN)}{Total\ Observations} \tag{4}$$

$$TPR = \frac{TP}{c_1} \tag{5}$$

$$FPR = \frac{FP}{\neg c_1} \tag{6}$$

According to values observed from confusion matrix, accuracy, TPR and FPR are calculated. Table 6 shows the observation of single as well as hybrid classifier evaluation according to various performance parameters.

Table 6. Single classifier and Hybrid model comparison

Classifier	DT	SVM	KNN	Hybrid Model
Accuracy	0.95	0.85	0.93	0.98
TPR	0.92	0.91	0.92	0.98
FPR	0.12	0.25	0.12	0.05

From Table 6 it is observed that decision tree classifier provides accuracy of 95% which is a good accuracy if data considered as a real time data. Limitation of this classifier is sometimes it generates biased output as DT cannot handle over fitting of data. In real time environment data is huge since sometimes DT does not provide this accuracy and provides more misclassifications. DT provides less false alarm rate when used with less traffic. Accuracy provided by SVM is only 85% as this classifier works only on linear data very efficiently. In terms of huge data coming from real time environment SVM fails in providing good accuracy misclassification goes on increasing. SVM also provides an overhead on the system to go for data normalization as string inputs cannot be handled by SVM. KNN classifier provides accuracy of 93% which is less than DT. This classifier fails to match the patterns of unknown attacks. It provides good accuracy when comes to the known attacks but fails for novel attacks. To remove the pitfalls of all single classifier and to improve accuracy and true positive rate as well as to decrease false positive rate hybrid model is proposed in this paper. This hybrid model provides accuracy of 98% in the real time environment. As this model makes use of ensemble approach using all mentioned classifier performance increases which can be observed form Table 6. Hybrid approach also obtained less false positive rate and good true positive rate. As compared to individual classifier hybrid model shows improvised performance. Figure 8 shows graphical representation of performance parameters for IDS.

Comparison of hybrid model with existing models is shown in Table 7. From observation it is observed that proposed model achieved better accuracy as compared to existing models now days. This model also provides less false positive rate and good true positive rate. This model uses ensemble approach which shows enhance performance than individual classifiers [26]. An approach which makes use of KNN algorithm to detect misuse as well as anomaly attack is elaborated by Guo et al. [27]. This approach has accuracy of 93.29% along with 0.78 as FPR. To improve accuracy in distributed and collaborative approaches various feature selection techniques are used. A dimension reduction technique chi-square is used by Thaseen et al. [28] to obtained accuracy of 95% with FPR 0.13. Dataset plays important role in system performance [29]. Clean data helps in increasing accuracy of SVM up to 94.71. But because of limitation of SVM this model provides high false alarm rate of 3.8. In [30] another hybrid approach is used which makes used or various strategies along with isolation algorithm to detect various types of attacks available in KDD dataset. Accuracy obtained is 95.1 with 3.0 as FAR [30].

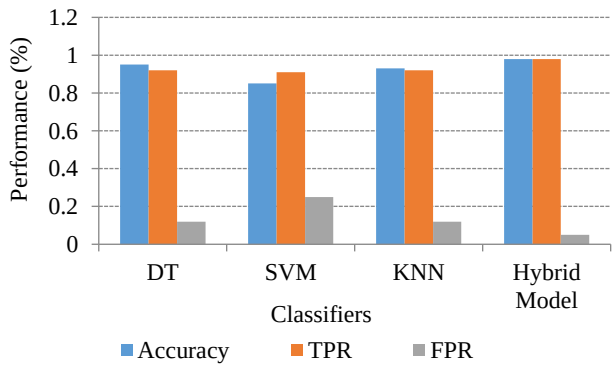


Figure 8. Performance parameters for IDS

To summaries as observed from results elaborated detection rate and accuracy of the ensemble classifier used in hybrid approach is more as compared to individual classifiers. Assembling improves the accuracy of the system as it helps in reducing the false alarm rate [31]. In proposed system central controller is used to distribute the packets and forward it in the network. These packets are checked for the malicious activities by central nide using firewall and then send further. This helps proposed system to work efficiently with good accuracy and less misclassification. Attacks can be of various types as attacks as collaborative and ca be detected with various algorithms [32, 33]. These attacks can be detected with multiclass SVM or unsupervised algorithms [34-36].

Table 7. Comparison of hybrid model with existing models

Models	Accuracy (%)	False Positive Rate
ELM-SVM [26]	95.86	2.13
Hybrid KNN [27]	93.29	0.78
Fusion chi-square and SVM [28]	95	0.13
Three tier IDS [29]	93.29	0.78
CSI-KNN [30]	95.1	3.0
Unsupervised classifiers [35]	94.5	2.7
One class SVM [36]	94.2	1.9
Proposed hybrid model	98	0.05

6. CONCLUSION

This paper presents a hybrid model which makes use of ensemble approach which helps in enhancing IDS performance. In this approach an algorithm which makes use of voting is used. This algorithm makes use of majority concept to enhance performance in terms of accuracy and detection rate. It also helps in increasing TPR and reducing FPR. Gini index dimensionality reduction technique is used to decrease number of features up to 21 out of 41 features of NSL-KDD dataset. Gini index feature selection technique is used to find importance of every feature. According to feature importance index features having highest values are selected for attack detection. Reduced number of features if used in distributed environment for real time traffic analysis it takes less processing time. Form results it has been observed that, accuracy of IDS progresses by 3% and rate of misclassification falls by 0.05 using reduced features. From comparison with existing IDS it can be concluded that as compared to existing model proposed hybrid system provides superior accuracy than any other existing IDS. Reduction in false alarm rate is also observed as compared to other existing IDS. Reduced false alarm rate tends towards less misclassification of packets. Ultimately it helps in increasing detection rate and accuracy of the IDS.

REFERENCES

[1] Scarfone K., Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS). National Institute of Standards & Technology, Gaithersburg, MD, United States, SP 800-94.

[2] Denning, D. (1987). An intrusion-detection model. IEEE Transaction on Software Engineering, 13(2): 222-232.

https://doi.org/10.1109/TSE.1987.232894
[3] Yan, Q., Yu, F. (2015). Distributed denial of service attacks in software-defined networking with cloud computing. IEEE Communication Magazine, 53(4): 52-59. https://doi.org/10.1109/MCOM.2015.7081075
[4] Depren, O., Topallar, M., Anarim, E., Ciliz, M. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Systems with Applications, 29(4): 713-722. https://doi.org/10.1016/j.eswa.2005.05.002
[5] Sung, A., Mukkamala, S. (2003). Identifying important features for intrusion detection using support vector machines and neural networks. Proceedings of the International Symposium on Applications and the Internet. IEEE Press, Orlando, Fla, USA, pp. 209-216. https://doi.org/10.1109/SAINT.2003.1183050
[6] Hasan, M., Nasser, M., Pal, B. (2013). On the kdd'99 dataset: Support vector machine based Intrusion Detection System (IDS) with different kernels. International Journal of Electronics Communication and Computer Engineering, 4(4): 1164-1170.
[7] Sindhu, S., Siva, S., Geetha, S., Kannan. (2012). A decision tree based light weight intrusion detection using A wrapper approach. Expert System and Applications, 39(1): 129-141. https://doi.org/10.1016/j.eswa.2011.06.013
[8] Wang, Q., Megalooikonomou, V. (2005). A clustering algorithm for intrusion detection. Defense and Security, International Society for Optics and Photoics, pp. 31-38. https://doi.org/10.1117/12.603567
[9] Pal, B., Hasan, M. (2012). Neural network & genetic algorithm based approach to network intrusion detection & comparative analysis of performance. 15th International Conference on Computer and Information Technology (ICCIT), Chittagong, pp. 150-154. https://doi.org/10.1109/ICCITechn.2012.6509809
[10] Barbara, D., Wu, N.N., Jajodia, S. (2001). Detecting novel network intrusions using Bayes. Proceedings of the 1st SIAM International Conference on Data Mining. https://doi.org/10.1137/1.9781611972719.28
[11] Hasan, M., Nasser, M., Pal, B., Ahmad, S. (2014). Support vector machine and random forest modelling for intrusion detection system. Journal of Intelligent Learning Systems and Applications, 6(1): 45-52. https://doi.org/10.4236/jilsa.2014.61005
[12] Jha, J., Ragha, L. (2013). Intrusion detection system using support vector machine. IJAIS Proceedings on International Conference and Workshop on Advanced Computing ICWAC, 3: 25-30. https://doi.org/10.5120/icwac1342
[13] Farnaaz, N., Jabbar, M. (2016). Random forest modelling for intrusion detection system. Elsevier Procedia Computer Science, 89: 213-127. https://doi.org/10.1016/j.procs.2016.06.047
[14] Choi, S., Chae, H., Jo, B., Park, T. (2013). Feature selection for intrusion detection and NSL-KDD. Recent Advances in Computer Science, 184-187.
[15] Roberto, P., Davide, A., Prahlad, F., Giorgio, G., Wenke, L. (2009). McPAD: A multiple classifier system for accurate payload-based anomaly detection. Elsevier Computer Network, 53(6): 864-881. https://doi.org/10.1016/j.comnet.2008.11.011
[16] Borji, A. (2007). Combining heterogeneous classifier for network intrusion detection. Advances in Computer Science. Computer and Network Security, Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 4046: 254-260. https://doi.org/10.1007/978-3-540-76929-3_24
[17] Amin, A., Reaz, M. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. Applied Soft Computing, 38: 360-372. https://doi.org/10.1016/j.asoc.2015.10.011
[18] Littlestone, N., Warmuth, M. (1994). The weighted majority algorithm. Elsevier Information and Computation, 108(2): 212-261. https://doi.org/10.1006/inco.1994.1009
[19] Giorgio, G., Roberto, P., Mauro, D., Fabio, R. (2008). Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Information Fusion, 9(1): 69-82. https://doi.org/10.1016/j.inffus.2006.10.002
[20] Jiong, Z., Zulkernine, M., Haque, A. (2008). Random-forests based network intrusion detection systems. IEEE Transaction of System, Man, Cybernetics, 38(5): 649-659. https://doi.org/10.1109/TSMCC.2008.923876
[21] Jungsuk, S., Takakura, H., Okabe, Y., Yongjin, K. (2009). Unsupervised anomaly detection based on clustering and multiple one-class SVM. IEICE Transaction on Communication, 92(6): 1981-1990. https://doi.org/10.1587/transcom.E92.B.1981
[22] Horng, S., Su, M., Chen, Y., Kao, T., Chen, R., Lai, J., Perkasa, C. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. Elsevier Expert Systems with Applications, 38(1): 306-313. https://doi.org/10.1016/j.eswa.2010.06.066
[23] Nguyen, H.H., Harni, N., Darmony, J. (2011). An efficient local region and clustering based ensemble system for intrusion detection. Proceeding of the 15th Symposium on International database engineering and Applications IDEAS '11. ACM, New York, USA, pp. 185-191. https://doi.org/10.1145/2076623.2076647
[24] Chou, T., Yen, K., Luo, J. (2008). Network intrusion detection design using feature selection of soft computing paradigms. International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2(11): 3722-3734. https://doi.org/10.5281/zenodo.1331229
[25] Hasan, M., Nasser, M., Ahmad, S., Molla, K. (2016). Feature selection for intrusion detection using random forest. Journal of Information Security, 7(3): 129-140. https://doi.org/10.4236/jis.2016.73009
[26] Al-Yaseen, W., Othman, Z., Nazri, Z. (2016). Real-time multi-agent system for an adaptive intrusion detection system. Elsevier Pattern Recognition Letters, 85: 56-64. https://doi.org/10.1016/j.patrec.2016.11.018
[27] Guo, C., Ping, Y., Liu, N., Luo, S. (2016). A two-level hybrid approach for intrusion detection. Elsevier Neurocomputing, 214: 391-400. https://doi.org/10.1016/j.neucom.2016.06.021
[28] Thaseen, I.S., Kumar, C.A. (2015). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University Computer and Information Sciences, pp. 1-11. https://doi.org/10.1016/j.jksuci.2015.12.004
[29] Hwang, T., Lee, T., Lee, Y. (2007). A three-tier IDS via data mining approach. Proceedings of Annual ACM Workshop on Mining Network Data, Minenet, pp. 1-6. https://doi.org/10.1145/1269880.1269882
[30] Kuang, L., Zulkernine, M. (2008). An anomaly intrusion

detection method using the CSI-KNN algorithm. ACM Symposium on Applied Computing, pp. 921–926. https://doi.org/10.1145/1363686.1363897

[31] Folino, G., Sabatino, P. (2016). Ensemble based collaborative and distributed intrusion detection systems: A survey. Elsevier Journal of Network and Computer Applications, 66: 1-16. https://doi.org/10.1016/j.jnca.2016.03.011

[32] Chen, Y., Hwang, K. (2006). Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. Journal of Parallel and Distributed Computing, 66(9): 1137-1151. https://doi.org/10.1016/j.jpdc.2006.04.007

[33] Perez, M., Tapiador, J., Clark, J., Perez, G., Gomez, A. (2014). Trustworthy placements: Improving quality and resilience in collaborative attack detection. Computer Networks, 58: 70-86. https://doi.org/10.1016/j.comnet.2013.08.026

[34] Zhou, C., Leckie, C., Karunasekera, S. (2010). A survey of coordinated attacks and collaborative intrusion detection. Computers and Security, 29(1): 124-140. https://doi.org/10.1016/j.cose.2009.06.008

[35] Song, J., Takakura, H., Okabe, Y., Nakao, K. (2013). Toward a more practical unsupervised anomaly detection system. Information Sciences, 231: 4-14. https://doi.org/10.1016/j.ins.2011.08.011

[36] Song, J., Hiroki, T., Yasuo, O., Kwon, Y. (2009). Unsupervised anomaly detection based on clustering and multiple one-class SVM. IEICE Transaction of Communication, 92(6): 1981-1990. https://doi.org/10.1587/transcom.E92.B.1981