International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

A Machine Learning Approach for Intrusion Detection using Ensemble Technique - A Survey

Shraddha Khonde*1, V. Ulagamuthalvi*2
*1Research Scholar, Department of CSE, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India
*2Professor, Department of CSE, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India

ABSTRACT

An Intrusion detection system is a machine or software that monitors the traffic in a network and on detection of a malicious packet, informs the user or a specific acting unit which can take further action and avoid the malicious packet from entering the network. In network intrusion, there may be multiple computing nodes attacked by intruders. The evidences of intrusions have to gather from all such attacked nodes. An intruder may move between multiple nodes in the network to conceal the origin of attack, or misuse some compromised hosts to launch the attack on other nodes. To detect such intrusion activities spread over the whole network, we present a new intrusion detection system (IDS) that classifies data with three different classifiers and an Ensemble technique that selects the majority of the three classifiers to assign the packet in the network as anomaly or normal. In this paper, we discuss a different ways to implement intelligent IDS, which classifies the normal traffic in a network with abnormal or attacked ones. This paper explains the method that used to generate such a system and the various classifiers used in the generation process. The dataset used to train classifiers can be NSL - KDD, KDD Cup 1999, KDD99 dataset. The IDS proposed here can serve many applications in the field of Military Systems, Banks and Social Networking websites where data is very sensitive. The paper also explains related work done in this field and briefly explains every classifier, the network attacks and the dataset.

General Terms: Network Security, Intrusion Detection, IDS, Artificial Intelligence, Machine Learning, Ensemble, SVM. Random Forest, Decision Tree, Collaborative IDS, Distributed IDS.

Keywords: IDS, Intrusion Detection System, Artificial Intelligence, AI, Majority Voting, Ensemble Learning, Random Forest, SVM, DT, Collaborative IDS and Distributed IDS.

I. INTRODUCTION

Intrusion has become a growing concern today. With the advent of new technologies each day and widespread of computers (from personal computers to embedded systems), security has become a very important issue. To name a few Attacks like Ransom ware, DoS, DDoS, U2R, R2L have become a great deal of concern to every computer in the network. Such attacks compromise the security of the computer and obtain access to sensitive data. Hence, Security of any network is a high priority issue that taken care. Various Intrusion Detection Systems (IDS) exist which help identify threats in the system but only an intelligent system will correctly yield them with maximum accuracy. With Data Mining, Machine Learning and Artificial Intelligence becoming pervasive in the computer world, it sets its foot into the area of Network Security as well. Hence, we could make full use of it and create a system that

could provide a secure environment for the users in a network. Aim is to create such a system that can resolve all the security issues related to web sites, personal computers and networks. One should aim is to create a Novel IDS which incorporates the methodologies of Data Mining, Machine Learning and Artificial Intelligence to identify the attacks in the network correctly with very less number of misclassifications which otherwise would go unidentified in traditional Intrusion Detection Systems. The following sections will discuss the technologies that be used to achieve better accuracy for detection.

A. Types of IDS

1. Signature-based IDS

Signature-based detection normally used for detecting known attacks. There are different definitions of attack signatures. In this paper, the main discussion will focus on content signatures, which represent a string of characters that appear in the payload of attack packets. No knowledge of normal traffic is required but a signature database needed for this type of detection systems. For worm detection, this type of system does not care how a worm finds the target, how it propagates itself or what transmission scheme it uses. The system looks at the payload and identify whether or not it contain a worm. One big challenge of signature-based IDS is that every signature requires an entry in the database, and so a complete database might contain hundreds or even thousands of entries. Each packet will be compare with all the entries in the database. This can be very resource- consuming and doing so will slow down the throughput and making the IDS vulnerable to Denial of Service attacks. Some of the IDS evasion tools use this vulnerability and flood the signature signature-based IDS systems with too many packets to the point that the IDS cannot keep up with the traffic, thus making the IDS time out and drop packets and as a result, possibly miss attacks [1]. Further, this type of IDS is still vulnerable against unknown attacks as it relies on the signatures currently in the database to detect attacks.

2. Anomaly-based IDS

The signature of a new attack is unknown before it is detection and carefully analyse. It is difficult to draw conclusions based on a small number of packets. In this case, anomaly-based systems detect abnormal behaviours and generate alarms based on the abnormal patterns in network traffic or application behaviours. Typical anomalous behaviours that may be captured include 1) misuse of network protocols such as overlapped IP fragments and running a standard protocol on a stealthy port; 2) uncharacteristic traffic patterns, such as more UDP packets compared to TCP ones, and 3) suspicious patterns in application payload. The big challenges of anomaly based detection systems are defining what a normal network behaviour is, deciding the threshold to trigger the alarm, and preventing false alarms. The users of the network are normally human, and people are hard to predict. If the normal model not defined carefully, there will be lot of false alarms and the detection system will suffer from degraded performance.

II. LITERATURE REVIEW

In [1] author gave in depth knowledge of the KDD-99 dataset .They separate the attack types into four types Basic, Content, Traffic and Host. The 2 main evaluation metrics, Detection Rate and False alarm rate .After clustering into 4 types using all attributes, 15 subsets were created. Each class dominance used to improve Detection rate and False alarm rate. The main aim was to find higher DR and decrease false alarm rate.

The more details explanation is in [2] where the system created an SVM classifier and the feature selection in the classifier done by using a combination K means and Information gain. Information gain gives us the importance of each feature .It is discovered that the difference between choosing top 23 features and top 30 makes a difference of difference of 0.05%.They first rank

features of based on information gain and then select features using K-means algorithm.

The author explain about various algorithms in [3] where they modelled a random forest classifier and compared it with j48 classifier. The modelled the classifier using NSL-KDD dataset. The dataset first clustered by using the classes of attacks after pre-processing. In pre-processing, for reducing features they used feature selection by finding out symmetrical uncertainty measure. In classifier training they created 100 trees .After classifier training, they compared the results using detection rate and false alarm rate. The accuracy was 99.67%, Detection rate was 99.83 and false alarm rate 0.00527%.The used 10 cross validation method.

In [4] they created IDS using SVM and Random Forest classifier. After comparing results they found out Random Forest is better in term of computational time. It is faster than SVM and it produces similar accuracy to SVM .They used radial basis kernel in this SVM and the accuracy for testing dataset is 92.99 and the Random Forest accuracy for testing in 91.41%.The precision for random forest was 10% more than even while the process time was less.

The analysis is done in [5] proved that in KDD dataset all the 41 features are not relevant by using, information gain, Gain ratio and Correlation based feature selection. The classifier used was decision tree classifier. They proposed a method AR (Attributed Ratio) which is a new method for giving importance of classes and compared with GR, IG and CFS. They proved that by using only 22 features out of 41 could achieve an accuracy of 99.79.

In [6][7] authors explained the specification that learns the normal ranges of values for each packet header field at the data link (Ethernet), network (IP), and transport/control layers (TCP, UDP, ICMP). PHAD detects some of the attacks in the ARPA data set that involve exploits at the transport layer and below.

The paper [8] suggests a method called pseudo-Bayes estimators as a means to estimate the prior and posterior probabilities of new attacks. Then a Naive Bayes classifier used to classify the instances into normal instances, known attacks and new attacks.

In [13] authors explains ensemble based SVM IDS. In this approach, the payload modelled using the technique of 2ν-grams is used. The payload represented as a sequence of 2-grams extracted at intervals of ν bytes. Authors vary the value of ν, and present the payload in different feature spaces obtained. Each feature space obtained has dimension 256. The authors adopt a pre-processing stage i.e., a clustering algorithm in order to reduce the dimension of the feature space. In the training phase, one model for represent payload done by ensemble of SVMs to model the normal traffic. In the detection phase, packet is analysed using ensemble SVM to output probability of normal traffic. The final probability decided by non-trainable combiner after combining output probabilities, of all ensemble SVM. Finally, the packet classified as normal if probability is above threshold. Authors use the well-known DARPA dataset for experiment.

In [14] number of classifier algorithms, i.e., ANN, SVM, the Decision Tree and k-NN are used. DARPA dataset used to compare using detection and false positive rates. All above mentioned classifiers are combined in an ensemble by means of a combination rule and the majority voting rule or the average rule and the "belief" function, which estimate the probabilities for a pattern belongs to the class or not. Experimental results gives more accurate results using ensemble-based approach as compared with the approach based on a single classifier.

In [15] novel approach used to describe work authors train two types of base classifiers, SVM-based and k-NN-based, on the KDD'99 dataset. In preliminary stage, the base classifiers are ensemble by a weighted majority voting. Weights chosen with three functions as particle swarm optimization (PSO),

another based on a variant of the PSO, which uses local unmoral and lastly the Weighted Majority Algorithm, introduced by Littlestone and Warmuth [16]. Ensemble technique provides improved results as compared with the base classifiers alone. An ensemble of neural networks is the basis of the architecture proposed in Sivatha Sindhuetal [17]. The Ada- Boost algorithm used to train an ensemble on the KDD'99 dataset. For feature selection, a genetic algorithm used to train the fitness function. The models based on decision tree built by running the C4.5 algorithm on the classes obtained from the above-described technique. Experimental results show that the accuracy obtained by ensemble is improves as compared to base classifier alone.

The paper [18] based on a modular ensemble. Each classifier contained in the ensemble used to classify traffic of specific service such as web service, mail service, and so on. Density-based solutions used, for testing the base classifiers. An approach based on the K-means clustering algorithm, v-SVC is used. Then, the base classifiers combined in the ensemble by using simple rules, i.e. maximum, minimum, and mean and the product rule. Testing done on the KDD'99 dataset prove that the most efficient approach is ensemble classifier.

An unsupervised IDS framework based on the random forest algorithm described in [19]. The system comprises a pre-processing phase that analyses feature selection for the network traffic used on dataset. Then, an analysis performed on random forest for offline data analysis. The random forest algorithm considers various bootstrap samples for creating regression tress in ensemble. When a packet/data needs to classified, each tree gives a vote and majority voting is consider for find exact class of data. In the detection stage, a number of outliers will found if it is exceed the threshold. Data set used KDD'99. Experiments prove that this method provides better and reduce false alarm rate.

Another unsupervised approach based on data mining techniques is described in [20] and in [21]. In the first paper, the overall architecture of IDS described, consists of three stages i.e., filtering, clustering and modelling. In the training phase, all attacks filtered and eliminated. This step accomplished by a data mining algorithm. After the data filtered, the system performs a clustering of the training data. Number of cluster is decided by user depend on accuracy parameter. Finally, in the modelling stage, for each cluster SVM trained. In the testing phase, the ensemble of SVMs used to detect normal traffic. In second paper, the authors extended the proposed method in order to tune the values of the above-mentioned parameters automatically without any intervention by the user. Number of clusters depend on the data and the architecture evaluated on real traffic. Both approaches compared and second approach is efficient in finding number of clusters for estimating the attack ratio as compared to normal traffic.

The approach explained in paper [22] uses BIRCH clustering algorithm. The KDD'99 dataset has five principal classes, Probe, Dos, U2R, R2Land normal traffic. BIRCH algorithm used to build feature trees, which give compact representation of dataset for each class. After feature selection SVMs are build, one for each attack class, trained and combined in an ensemble together for testing. Compared to decision trees and K-means classifiers, SVM shows better performance.

In the approach proposed by author in [23], data set is prepared by collecting data from sources like operating system audits, network packets and system logs. The domain expert knowledge used to label dataset, and to constitute the IDS training set, which split into a training and a validation set. Then, K-means algorithm, used to detect the normal behaviour. A decision tree algorithm used to build an ensemble of different classifiers. To combine output of various classifiers output a weighted mean is used. Experiments conducted on the dataset KDD'99 and it

is prove that the system use bagging and boosting methods to ensemble classifiers to improve accuracy.

III. CLASSIFIERS

A. Decision Tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. These are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. In decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values (or expected utility) of competing alternatives are calculated. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process repeated on each derived subset in a recursive manner called recursive partitioning. Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items. Different algorithms use different metrics for measuring "best". These generally measure the homogeneity of the target variable within the subsets.

Types of nodes:

1. Decision nodes - typically represented by squares
2. Chance nodes - typically represented by circles
3. End nodes - typically represented by triangles

Decision Tree Elements

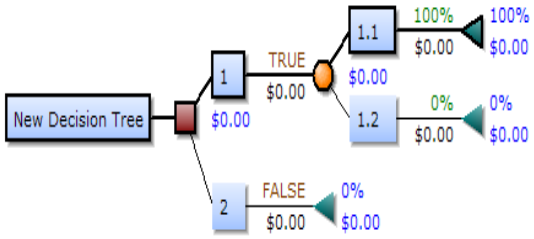

Figure 1. Decision Tree

B. Random Forest (RF)

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.Random forest is an ensemble classifier. It has a higher classification accuracy compared to single decision tree. Random forest contains many decision trees, which trained with the same dataset and different features, selected at random. It avoids overfitting as features and data randomly selected.

Training:

The goal of Random Forest is to use distributed approach for classification. The features used to train decision trees will selected using their importance in the KDD-99 dataset. The trained model could place on a distributed network to increase real time performance and reliability.

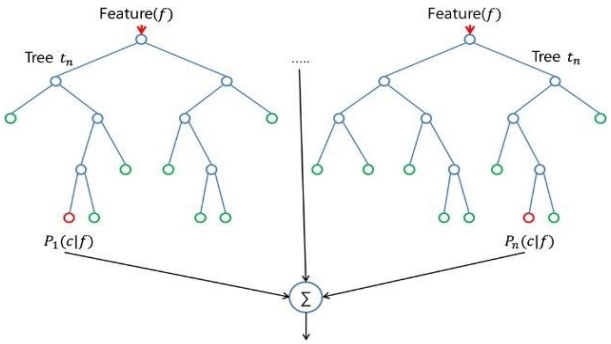

Figure 2. Random Forest Tree

The class assigned by the decision trees would put to a vote and majority class will be assigned to the input data

$$P(c|f) = \sum_{1}^{n} P_n(c|f)$$

The number of trees in the forest will depend on the current network traffic and processing capabilities available. Maximum idle processing power could used to increase number of trees in the forest will give classification that is more precise. Advantage of Random Forest is it work accurately on collaborative as well as distributed network.

C. Support Vector Machine (SVM)

In SVM classifier training, we create 'n' number of hyper planes for 'n' number of classification. SVM could use for both, classification and regression but in a case we will be using for classification. Hyper plane is a plane which divides two different classes of nodes in a space and hence classification is done.

Steps for Generating SVM:

a) Data-Pre-processing:

SVM's are incapable of processing categorical data since they only process numerical data. In order to train SVM from KDD dataset we to need to convert string data into appropriate numerical data for training the classifier. We also need to save the process of conversation in order to test the live data because classifier will not work if live data not converted according to the conversation process of training data.

Steps:

1) Scan string value.
2) Check if numerical value assign.
If assigned replace string with numerical value.
3) If not assigned assign value and replace string.
4) Save the replacing values for future use.

b) Data Normalization:

Data normalization is extremely important for training classifier using KDD dataset as range of value for each feature varies a lot. If data is not normalized, then it may occur that the trained classifier would be biased to certain features only and also training time increases and the accuracy decreases and also the value with which we are dividing our data must be saved in order divide the live data to classify it.

$$N_2 = (N_1 * \min) / (\max - \min)$$

where, N_2 = New Value

N_1 = Old Value

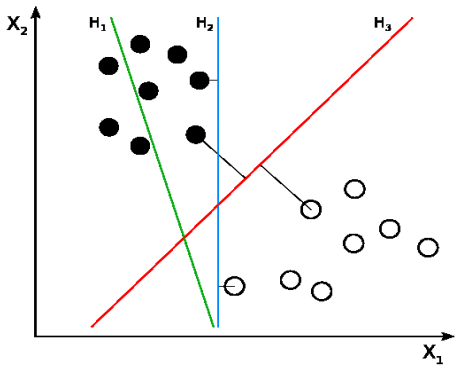

Figure 3. Hyperplane representation of SVM

IV. ENSEMBLE LEARNING

Ensemble learning is a method to improve the accuracy of a model by creating a set of several Machine Learning Classifiers and then predicting the most fitting class of the input data. This could done by making use of a majority-voting scheme between all classifiers and selecting the class of the data, which has most votes that is the class which majority of the classifiers have predicted. Ensemble Techniques could Bagging, Boosting or Bayes Optimal Classifier.

We are making use of Boosting. Boosting attaches weights to each data. One potential drawback of boosting is overfitting.

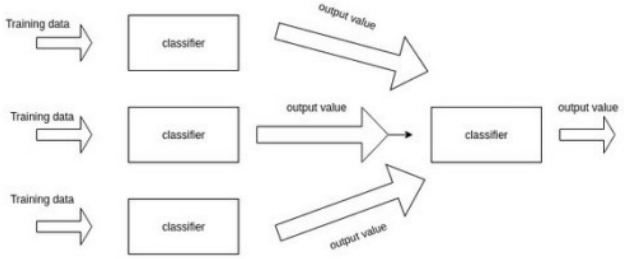

Figure 4. Transition of an Ensemble Method

In the literature, we can analyse a wide variety of anomaly detection systems based on various machine-learning techniques. Many studies have implemented single-stage learning algorithms, such as artificial neural networks (ANN), genetic algorithms (GA) and support vector machines (SVM). However, systems based on a combination of several methods, such as hybrid or ensemble systems, have been common as well. This section presents an overview of such approaches for intrusion detection systems.

Early research by Dietterich [9] showed, that ensembles are better than single-component classifiers, in terms of classification accuracy. Error rate is reduce using ensemble of multiple classifier. The advantages of ensemble classifiers is more for intrusion detection. As in growing era different types of intrusions are available more detectors will be needed for detection Axelsson [10] if one base classifier fails to classify an attack, then another classifier can classify it. Based on an ensemble's structure, two general approaches might distinguished: (*i*) homogeneous ensembles, where all classifiers use same technique to generate classifier and (*ii*) heterogeneous ensembles, which utilize different base classifiers. Bagging and boosting are the ensemble techniques often used to generate homogeneous ensembles and heterogeneous ensembles can be form using stacking and voting.

A. Homogeneous ensembles for IDS

In general, homogeneous ensembles could view as a simple and effective way of extending the classification hypotheses of a single classification algorithm by creating several variations of that classifier. Although there are numerous ensemble methods by which this could be achieve, the core principles are the same: the aggregation of several relatively simple decision rules should lead to a more sophisticated and reliable final decision. Usually, the selected classifier trained with different training subsets, at various stages of ensemble development. As a result, the classifier analyses the problem from different perspectives, and, each time, aggregates the knowledge gained towards the definition of an ensemble classification hypothesis. This method has the advantage of solving the computing time limitation, but cannot used in real-time.

B. Heterogeneous ensembles for IDS

The defining characteristic of heterogeneous ensembles is that the final decision based on the classification rules of diverse base classifiers. The chief obstacle to creating such ensembles is that each expert in the ensemble employs a particular method to construct its classification hypothesis. To generate heterogeneous ensembles, the output of each base classifier must be interpretable in the same way. There are various strategies for aggregating the classification results into a final decision, and the voting procedure is one of the simplest and easiest methods to implement. In this section, an overview of heterogeneous ensemble classifiers presented, with particular attention given to methods based on voting and weighted voting strategies.

Meng and Kwok [11] experimented with both single and ensemble classifiers composed of J45, *k*NN, and SVM, for classification of the 1998 DARPA data set. They analyse that an ensemble of all three classifiers, use majority voting for getting better performance. Ensemble method gives the improved performance as compare to single classifier. In heterogeneous ensemble methods majority voting used for pre-processing to improve accuracy. The variety of classification approaches explored is increased. Detection of attacks, reduction of false alarms, and reduction of response times are the better result obtained by using ensemble classifier.

V. METHODS FOR CREATING ENSEMBLE CLASSIFIERS

In recent years, an abundance of ensemble-based classifiers been produced and improved. Nonetheless, a number of these classifiers are variations on just a few well-established algorithms with capabilities that

comprehensively validated and broadly published. An overview of the most commonly used ensemble algorithms presented in this section.

A. Bagging

Breiman's bootstrap aggregating method, or "bagging" for short, was one of the first ensemble-based algorithms, and it is one of the most natural and straightforward ways of achieving a high efficiency [9]. In bagging, a variety of results produced, using bootstrapped copies of the training data; that is, numerous subsets of data randomly drawn with replacement from the complete training data. A distinct classifier of the same category modelled, using a subset of the training data. Fusing of particular classifiers achieved by the use of a majority vote on their selections. Thus, for any example input, the ensemble's decision is the class selected by the greatest number of classifiers. An approach that derived from bagging called the "random forests" classifier. It received its name because it builds a model from a number of decision trees [9]. A decision trees used for detection. As in bagging, the parameters can be bootstrapped copies of the training data; however, in contrast with bagging, they also can be particular feature subsets, which is the practice in the random subspace method.

Another approach that derived from bagging called "pasting of small votes." Unlike bagging, pasting small votes was an approach devised to operate on large data sets [9]. Data sets of a large size partition into subsets of a smaller size, which called bites and those bites, used to train different classifiers. Pasting small votes has led to the creation of two variations: the first one, known as Rvotes, generates the data subsets at random; the other, called Ivotes, builds successive data sets, considering the relevance of the instances. Of the two, Ivotes show to yield better outcomes [12], similar to the idea present in the boosting-based methods, by which each classifier directs the most relevant instances for the ensemble part that is in use.

B. Boosting

Schapire, showed it in 1990, that a weak learner, namely an algorithm that produces classifiers that can slightly outperform random guessing, can be transformed into a strong learner, namely an algorithm that constructs classifiers capable of correctly classifying all of the instances except for an arbitrarily small fraction. Boosting generates an ensemble of classifiers, as does bagging, by carrying out resampling of the data and combining decisions using a majority vote. However, that is the extent of the similarities with bagging. Re-sampling in boosting carefully devised as to supply consecutive classifiers with the most informative training data. Essentially, boosting generates three classifiers as follows: A random subset of the available training data used for constructing the first classifier. The most informative subset given for the first classifier used for training the second classifier, where the most informative subset consists of training data instances, such that the first classifier and the other half correctly classified half of they were misclassified. Finally, training data for the third classifier are made of instances on which the first and second classifiers were in disagreement. A three-way majority vote used, to combine the decisions of the three classifiers. In 1997, Freund and Schapire presented a generalized version of the original boosting algorithm called "adaptive boosting" or "AdaBoost" for short. The method received that name from to its ability to adapt to errors related to weak hypotheses, which obtained from Weak Learn. AdaBoost.M1 and AdaBoost.R are two of the most frequently used variations of this category of algorithms, because they are suitable for dealing with multi-class and regression problems, respectively.

AdaBoost produces a set of hypotheses, and then uses weighted majority voting of the classes determined by the particular hypotheses in order to combine decisions. A weak classifier trained to generate the hypotheses, by drawing instances from a successively refreshed distribution of the training data. The updating of the distribution guarantees that it will be

more likely to include in the data set for training the subsequent classifier examples that wrongly classified by the preceding classifier. Thus, the training data of successive classifiers tend to advance toward increasingly hard-to-classify instances.

VI. METHODS THAT COMBINE CLASSIFIERS

The practice of combining classifiers is the second fundamental element present in ensemble schemes. This approach uses combination rules that usually categorized according to the following criteria: (i) combination rules that are trainable vs. those that are non-trainable; or, alternatively, (ii) class labels vs. class-specific applicable combination rules. An independent algorithm establishes the parameters required by the combiner, which commonly called "weights," in the case of trainable combination rules. An example of this category of methods is the EM algorithm used in the mixture of competing expert's model. In the trainable combination rules, the parameters are generally instance-specific, and known as dynamic combination rules. In contrast, in the case of non-trainable combination rules, the training is dependent; instead, it incorporated to the training of the ensembles. Weighted majority voting falls into this category of non-trainable rules, as discussed below, given that the weights directly obtained when the classifiers created. According to the other taxonomy, class labels having applicable rules that solely require the classification decision opposed to those having inputs consisting of continuous-valued outputs produced by particular classifiers.

Generally, what these values represent is to what extent the classifiers support each class, and, consequently, they can be used to estimate class-conditional posterior probabilities $P(\omega j|x)$. Two conditions are required for that last statement: (i) the values have to properly normalized, so that they add up to one considering all classes; and (ii) the training data used by the classifiers are required to be sufficiently dense. Those two models produce continuous-valued outputs that commonly used as posterior probabilities, although the second required condition concerning sufficiently dense training data often not met.

VII. DATASET

A. Darpa 98

The DARPA 98 (Defense Advanced Research Projects Agency) was the first dataset, which used for detection of attacks in IDS. This dataset consist of network connections raw data organized as records .It includes a training set around 5 million connection records and a testing set around 2 million connection records. Each connection record labeled as normal connection or specific attack type. The attack types fall in one of the four categories: Denial of Service attack (DoS), User to Root (U2R), Remote to Local (R2L) and Probe.

B. Kdd

KDD 99(Knowledge Discovery and Data Mining) is a processed version of DARPA 98 dataset where each connection record represented by a vector of 41 features and labelled as normal connection or attack type. KDD 99 dataset is the most reliable dataset for network security. Most of datasets used for network security are derived from this dataset such as NSL-KDD, Corrected-KDD, 10% KDD etc. It has over 300 thousands of entries and all of them labeled .Even though the data set created in 1999; it is the most preferred dataset. There are four main species of attacks, around 35 subspecies, and covering most of the network related attacks.

VIII. CONCLUSION

This paper, mainly reviews the various data mining and machine learning classifiers approach for network intrusion detection suitable to take advantage of modern parallel/distributed and cloud environments. We also describe an important technique called Ensemble Learning, which if used for attack detection, decreases the misclassification rate, significantly. We also discussed other related work in literature survey done on similar grounds.

Various measures could take to improve the detection rate with the help of better machine learning algorithms, which take less processing time. To get more results we replace Bagging and Boosting by a better Ensemble Technique where all the above-mentioned classifiers will be working together as a single classifier. The decision for normal and malicious activities/data would take based on majority voting algorithm. With ensemble method, we are reducing overfitting, which takes place in boosting. Ensemble techniques gives more accurate detection rate as compared to single classifier technique. To improve accuracy and reduce false alarm rate we are using Ensemble rather than single classifier for distributed network IDS.

IX. ACKNOWLWDGMENTS

My sincere thanks to Dr. V. Ulagamuthalvi for helping me in completion of this paper and inculcating her knowledge in this domain with me. Her constant motivation and support at every stage of development has helped me design this paper. My special thanks to all faculties of Department of Computer Engineering, Sathyabama Institute of Science and Technology for their constant support.

X. REFERENCES

[1]. Preeti Aggarwala, Sudhir Kumar Sharma "Analysis of KDD Dataset Attributes" ICRTC 2015.

[2]. Jayshree Jha and Leena Ragha". Intrusion Detection System using Support Vector Machine" ICWAI 2013.

[3]. Nabila Farnaaz and M. A. Jabbar,"Random Forest Modeling for Intrusion Detection System" Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd. IMCIP 2016.

[4]. Md. Al Mehedi Hasan, Md Nasser, Biprodip Pal and Shamim Ahmad," Support Vector Machine and Random Forest Modeling for Intrusion Detection System Forman", G. 2003. An extensive empirical study of feature selection metrics for text classification. JILSA 2014 (Mar. 2003), 1289-1305.

[5]. Sang-Hyun Choi, Hee-su Chae, Byung-oh Jo and, Twae-kyung Park, "Feature Selection for Intrusion Detection and NSL-KDD".

[6]. M. Mahoney and P. Chan, "PHAD: Packet header anomaly detection for identifying hostile network traffic", Technical report, Florida Tech., technical report CS-2001-4, April 2001.

[7]. Mahoney M. and P. Chan, "Learning models of network traffic for detecting novel attacks", Technical report, Florida Tech 2002.

[8]. D. Barbara, N. Wu and S. Jajodia, "Detecting Novel Network Intrusions using Bayes Estimators", Proceedings of the 1st SIAM International Conference on Data Mining, 2001.

[9]. Dietterich TG." Ensemble methods in machine learning.In: Multiple classifier systems".

[10]. Axelsson S. "Intrusion detection systems: a survey and taxonomy", Tech. rep., Technical report Chalmers University of Technology, Goteborg, Sweden; 2000.

[11]. Meng Y, Kwok L-F. "Enhancing false alarm reduction using voted ensemble selection in intrusion detection. Int J Computer IntellSystem 2013;6(4):626-38.

[12]. Chawla NV, Hall LO, Bowyer KW, Moore T Jr, Kegelmeyer WP, "Distributed pasting of small votes. In: International workshop on multiple classifier systems. Springer; 2002. p. 52-61.

[13]. Perdisci Roberto, Ariu Davide, Fogla Prahlad, Giacinto Giorgio, Lee Wenke. McPAD "A multiple classifier system for accurate payload-based anomaly detection". Computer Network.2009;53(6):864-81

[14]. Borji, Ali, "Combining heterogeneous classifiers for network intrusion detection". In:Cervesato, Iliano,(Ed.), Advances in Computer Science-ASIAN2007. Computer and Network Security, Lecture Notes in Computer Science, vol.4846. Springer, Berlin,Heidelberg,pp.254-260, 2007.

[15]. Aburomman Abdulla Amin, Ibne Reaz Mamun Bin. "A novel SVM-kNN-PSO ensemble method for intrusion detection system" Applied .Soft Computing. 2016;38:360-72.

[16]. Littlestone Nick, Warmuth Manfred K."The weighted majority algorithm" .International Conference on Computing 1994; 108(February (2)):212-61

[17]. Sivatha Sindhu Siva S, Geetha S, Kannan "A. Decision tree based lightweight intrusion

detection using a wrapper approach". Expert System .Applications.2012;39(1): 129-141.

[18]. Giacinto Giorgio, Perdisci Roberto, DelRio Mauro, Roli Fabio. "Intrusion detection in computer networks by a modular ensemble of one-class classifiers". International Conference on Fusion 2008;9(1):69-82 Special Issue on Applications of Ensemble Methods.

[19]. Zhang Jiong, Zulkernine M, Haque "A. Random-forests based network intrusion detection systems" IEEE Transaction of System Man Cybern. PartC:Appl. Rev.2008;38 (September (5)):649-59

[20]. Jungsuk Song, Takakura Hiroki, Okabe Yasuo, Yongjin Kwon. "Unsupervised anomaly detection based on clustering and multiple one-class SVM" IEICE Trans.Com- mun. 2009;92(6):1981-90.

[21]. Song Jungsuk, Takakura Hiroki,Okabe Yasuo,Nakao Koji. "Toward a more practical unsupervised anomaly detection system".Inf.Sci.2013;231(0):4-14 Data Mining for Information Security.

[22]. Horng Shi-Jinn, SuMing- Yang, Chen Yuan-Hsin, Kao Tzong -Wann, ChenRong-Jian, Lai Jui-Lin DwiPerkasa Citra. "A novel intrusion detection system based on hierarchical clustering and support vector machines". ExpertSyst.Appl.2011;38 (1):306-13

[23]. Nguyen, Hoa Huu, Harbi, Nouria, Darmont, Jérôme, 2011. "An efficient local region and clustering-based ensemble system for intrusion detection". In: Proceedings of the15th Symposium on International Database Engineering & Applications, IDEAS '11, ACM ,NewYork, NY, USA, pp. 185-191.