SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.592

SPECIAL ISSUE PAPER

Hybrid intelligent systems for detecting network intrusions

Mrutyunjaya Panda[1]*, Ajith Abraham[2,3] and Manas Ranjan Patra[4]

[1] Department of ECE, Gandhi Institute for Technological Advancement, Bhubaneswar-54, India
[2] Machine Intelligence Research Laboratory, (MIR Labs), Scientific Network for Innovation and Research Excellence, WA, USA
[3] VSB-Technical University of Ostrava, Ostrava-Poruba, Czech Republic
[4] Department of Computer Science, Berhampur University, Berhampur, India

ABSTRACT

This paper intends to develop some novel hybrid intelligent systems by combining naïve Bayes with decision trees (NBDT) and by combining non-nested generalized exemplar (NNge) and extended repeated incremental pruning (JRip) rule-based classifiers (NNJR) to construct a multiple classifier system to efficiently detect network intrusions. We also use ensemble design using AdaBoost to enhance the detection rate of the proposed hybrid system. Further, to have a better overall detection, we propose to combine farthest first traversal (FFT) clustering with classification techniques to obtain another two hybrid methods such as DTFF (DT + FFT) and FFNN (NNge + FFT). Finally, we use Bayesian belief network with Tabu search combined with NNge for better detection rate. Because most of the anomaly detection uses binary labels, that is, anomaly or normal, without discussing more details about the attack types, we perform two-class classification for our proposed methodologies in this paper. Substantial experiments are conducted using NSL-KDD dataset, which is a modified version of KDD99 intrusion dataset. Finally, empirical results with a detailed analysis for all the approaches show that hybrid classification with clustering DTFF provides the best anomaly detection rate among all others. Copyright © 2012 John Wiley & Sons, Ltd.

KEYWORDS

intrusion detection; hybrid intelligent systems; Bayesian networks; decision trees; rule-based classifiers; clustering

***Correspondence**

Mrutyunjaya Panda, Department of ECE, Gandhi Institute for Technological Advancement, Bhubaneswar-54, India.
E-mail: mrutyunjaya.2007@rediffmail.com

1. INTRODUCTION

Intrusion is defined to be the set of actions that attempt to compromise the proprietary business plans (integrity and confidentiality) or loss of critical business data and disruption of services (availability) of system resources [1]. An intrusion detection system (IDS) is a system for detecting such intrusions and thus works as the last defensive mechanism in system security [2]. There are basically two types of IDS, namely host-based IDS (HIDS) and network-based IDS (NIDS). Whereas HIDS operates on information collected from within an individual computer system such as system logs and audit trails [3,4], NIDS performs packet logging, real-time traffic analysis of IP networks and tries to discover if intrusion occurs [5,6]. Further, IDS can be categorized into anomaly detection and misuse detection systems [7]. Anomaly detection systems detect attacks by observing deviations from the normal activities of the system. Misuse detection systems, on the other hand, detect known attacks by using predefined attack patterns and signatures.

Although there has been an extensive body of work in this field, particularly in the domain of data mining by using KDDCup 1999 benchmark dataset [8], recently, McHugh found KDD99 dataset to have some inherent problems [9]. The new version of KDD dataset, NSL-KDD [10], is publicly available for researchers, by considering some issues addressed in [9] to model their IDS. In this paper, we propose to use some novel hybrid data-mining methods by using NSL-KDD dataset with two-class classification to model an efficient anomaly-based network intrusion detection system. The organization of this paper is as follows: Section 2 briefly introduces the related overview of the work carried out so far in this area of research. Section 3 presents the details about the KDDCup 1999 and NSL-KDD datasets for training and testing the proposed methodology. Some theoretical background about the naïve Bayes (NB), decision trees (DT), rule-based classifiers, and Farthest first clustering are provided in Section 4. The proposed methodology used in this paper is discussed in Section 5 followed by the experimental results and discussion in Section 6. Finally, we conclude our work in Section 7.

2. RELATED WORK

Dominik Fisch *et al.* [11] have demonstrated a case study by comparing the classification abilities of radial basis function

classifiers with multilayer perceptrons, the neuro-fuzzy systems, DTs, fuzzy k-means, and nearest neighbor classifiers in detecting network intrusions. In this, the authors concluded that radial basis function classifiers are found suitable in identifying the novel attacks. Zanial *et al*. [12] proposed an ensemble of one class classifier where each adopts different learning algorithms such as the following: linear genetic programming, random forest, and adaptive neuro-fuzzy inference system to build an NIDS. Mukkamala *et al*. [13] proposed three variants of neural networks, support vector machine (SVM), and MARS as components in their IDS and demonstrated to obtain better performance in comparison with single classifier approach. Panda and Patra [14] introduced a hybrid approach by combining NB with decision tables to enhance the performance of the IDS. In [15], the authors used the hybrid clustering approach by combining COBWEB and farthest first traversal (FFT) clustering to detect network intrusions that fall in rare attack categories. A novel intrusion detection method using probabilistic neural network and adaptive boosting is proposed in [16]. In this, the authors integrated an adaptive boosting technique and a semi-parametric neural network to obtain good trade-off between accuracy and generality to build an NIDS. In [17], Farid *et al*. combined NB with DT (ID3) for adaptive intrusion detection. Panda and Patra [18] modeled an ensembling rule-based classifiers for detecting network intrusions by using decision tables, non-nested generalized exemplars (NNge), extended repeated incremental pruning (JRip), and ripple down rules. A hybrid artificial immune system and self-organizing map for network intrusion detection is proposed by Powers and He in [19]. Shon and Moon [20] proposed a hybrid machine-learning approach to network anomaly detection by using enhanced SVM with *m*-fold cross-validation obtaining an average detection rate of 87.74% with 10.2% false positive rate. All the aforementioned papers use the KDDCup 1999 benchmark dataset to build an efficient network intrusion detection system, but as discussed in [10], KDDCup 99 dataset poses some inherent problems supported by Mahoney and Chan [21], who analyzed DARPA background network traffic and found evidence of simulation artifacts that could result in an over estimation of the performance of some anomaly-based NIDS; recently, few researchers in [10,22] concentrate on using NSL-KDD dataset, in place of the earlier one in this area of research. Kou *et al*. [23] proposed a multicriteria mathematical programming model for multiclass classification for network intrusion detection by using KDD 1999 data and NeWT data collected from STEAL Lab [24]. None of the aforementioned works have proposed a novel approach on NSL-KDD data for enhancing the performance of the IDS, which we tried to perform using multiple classifier or hybrid approach in this paper.

3. INTRUSION DATASET

In this paper, we discuss about the two types of benchmark dataset used to detect network intrusions. They are as follows:

- KDDCup 1999 intrusion dataset
- NSL-KDD dataset

3.1. KDDCup 1999 intrusion dataset

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs with an objective to evaluate research in intrusion detection and to set up an environment to acquire 9 weeks of the raw TCPDump data for a LAN simulated in a typical US Air Force environment peppered with multiple attacks [25]. The 1999 KDDCup intrusion data are a subset of this dataset.

The raw training data were about 4 GB of compressed binary TCPDump data collected from 7 weeks of network traffic, which was further processed to obtain about five million connection records. Similarly, the 2 weeks of test data yielded around two million connection records. Each connection is labeled either as normal or as an anomaly (or attack) of a specific type that falls under the following attack categories:

- Probing: It is a type of attack where an attacker scans a network to gather information to find known vulnerabilities. This type is most common as it requires very little technical expertise.
- DoS: The denial of service (DoS) attack occurs when an intruder makes some computing or memory resource too busy or too full to handle legitimate requests.
- U2R: User to root (U2R) is a class of attacks where an intruder tries to access through a normal user account on the system by gaining root access.
- R2L: In remote to local (R2L) attack, the attacker sends packets to a machine over a network that exploits the machine's vulnerability to gain local access as a user illegally.

Although most researchers use KDDCup 1999 dataset for designing their intrusion detection system by using machine-learning approaches such as neural network, fuzzy logic, Bayesian learning, genetic algorithms, and variants of SVMs, recently, it is criticized by McHugh [9] mainly because of the characteristics of the synthetic data. One among many inherent problems found in the KDD 1999 dataset is that TCPDump, which is used as traffic collectors in DARPA 1998, is most often getting overloaded and therefore drop packets in heavy traffic load without considering the possibility of the dropped packets. Therefore, the KDD 1999 dataset is found unsuitable nowadays to model an NIDS because of the following deficiencies in KDD 1999 dataset:

- Redundant records: The most important deficiencies in the existing KDD1999 dataset are the huge number of redundant records, which causes the learning algorithms to be biased towards frequent records and thus prevents them from learning unseen records that fall under rare attack categories (U2R and R2L attack types) that are usually found more harmful to computer

networks. Further, the evaluation of the NIDS tends to be biased because of the existence of these redundant records in the test dataset.
- Level of difficulty: It has been observed from the results obtained by many researchers over KDD1999 dataset to attempt to devise a complex IDS produces high accuracy rates with KDD full training set and randomly selected testing data from KDD test set. It is argued that these methods on the KDD 1999 dataset are not appropriate actions.

3.2. NSL-KDD intrusion dataset

The new version of KDD dataset, NSL-KDD, is publicly available for researchers through the website developed by Tavallaee *et al.* [10] in their detailed analysis of the KDDCup 1999 dataset. It is reported further by the authors that even though the new dataset still suffers from some of the problems as discussed in [9] and may not be a perfect representative of the real networks because of the lack of publicly available datasets for NIDS, still this can be applied as an effective benchmark dataset to design an efficient IDS.

The NSL-KDD dataset does not contain the redundant records in both train and test datasets. The generated datasets, NSL-KDD Train+ and NSL-KDD Test+ included 125,973 and 22,544 connection records, respectively. For experimental purposes, we employed the first 20% of the records in NSL-KDD Train+ as the training data.

Further, it is also reported that the original KDD 1999 testing dataset is skewed and unproportionately distributed, which makes it unsuitable for testing NIDS. Therefore, the performance evaluations obtained by using many machine-learning algorithms are unreliable and cannot be considered as good indicators of building an efficient NIDS, so we use NSL-KDD Test+ dataset for the evaluation of our proposed methodology.

4. DATA-MINING APPROACHES FOR DETECTING NETWORK INTRUSIONS

In this section, we briefly outline about some data-mining approaches applied in network intrusion detection systems. Data mining is considered as an attempt to extract knowledge in the form models from data, which may not be realized easily with the naked eye. Although data-mining techniques include classification, regression, clustering, association rule analysis and so on, intrusion detection can be thought of as a classification problem so as to classify the records either normal or intrusive without having much insight about the various attack types.

4.1. Decision trees

These are powerful and popular tools for classification and prediction. The attractiveness of tree-based methods is largely due to the fact that in contrast to neural networks, DTs represent rules. A DT is a tree that has three main components: nodes, arcs, and leaves. Each node is labeled with a feature attribute that is the most informative among the attributes not yet considered in the path from the root; each arc out of a node is labeled with a feature value for the node's feature, and each leaf is labeled with a category or class. A DT can then be used to classify a data point by starting at the root of the tree and moving through it until a leaf node is reached. The leaf node would then provide the classification of the data point.

Decision tree induction algorithm [26] works recursively to learn knowledge on classification. The following steps are considered for the DT induction:

(1) At first, root node is selected for an attribute, which must effectively split the data for efficient tree construction.
(2) Then, each data split attempts to reduce the set of instances available in the actual data until same classification for all is achieved.
(3) Information gain for each split is now calculated to see how the randomness is removed while constructing tree step wise.
(4) The split having the most information gain is considered to be the best split.
(5) The attribute with most information gain is now chosen as root node and continues the calculation recursively till the final data classification step is achieved.

The DT induction with the aforementioned approach has a number of possible shortcomings. One common issue arises when an attribute has a large number of uniquely identifying values. An example of this could be social security numbers or other types of personal identification numbers. In this case, there is an artificially high decision value to the information, where the ID classifies each and every person and distorts the algorithm by overfitting the data. One solution is to use an information gain ratio that biases attributes with large numbers of distinct values.

To illustrate the post-pruning of the rules, let us consider the following rule generated from the tree:

$$\text{IF}(service{=}login)\text{^}(flag{=}SF)\ \text{THEN}\ class{=}ftp\_write \quad (1)$$

This rule is pruned by removing any antecedent whose removal does not worsen its estimated accuracy. The pruning algorithm is based on a pessimistic estimate of the error rate associated with a set of N cases, out of which E number of cases do not belong to the most frequent class. Instead of E/N, J48 algorithm determines the upper limit of the binomial probability when E events have been observed in N trials, using a user-specified confidence whose default value is 0.25.

It can also be seen that J48 rules have interesting properties for the intrusion detection because it generates good generalization accuracy. New intrusions may appear after the building process whose forms are quite similar to known

attacks that are considered a priori. By using the generalized accuracy of the rules, new attack variations could then be detected using different rules. Real-time IDSs require short rules for efficiency. Therefore, post-pruning of rules can generate accurate conditions and can avoid over fitting. This improves the execution time for real-time intrusion detection.

4.2. Bayesian networks

A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. When used in conjunction with statistical techniques, Bayesian networks have several advantages for data analysis [27]. Firstly, because Bayesian networks encode the interdependencies between variables, they can handle situations where data are missing. Secondly, Bayesian networks have the ability to represent causal relationships. Therefore, they can be used to predict the consequences of an action. Lastly, because Bayesian networks have both causal and probabilistic relationships, they can be used to model problems where there is a need to combine prior knowledge with data. Several researchers have adapted ideas from Bayesian statistics to create models for anomaly detection [28–30]. The Bayesian network is restricted network that has only two layers and assumes complete independence between the information nodes (i.e., the random variables that can be observed and measured). These limitations result in a tree-shaped network with a single hypothesis node (root node) that has arrows pointing to a number of information nodes (child node). All child nodes have exactly one parent node, that is, the root node, and no other causal relationship between nodes is permitted.

The naïve Bayesian networks have some disadvantages. First, as pointed out in [28], the classification capability of naïve Bayesian networks is identical to a threshold-based system that computes the sum of the outputs obtained from the child nodes. Secondly, because the child nodes do not interact between themselves and their output only influences the probability of the root node, incorporating additional information becomes difficult as the variables that contain the information cannot directly interact with the child nodes.

Another area, within the domain of anomaly detection, where Bayesian networks have been frequently used is the classification and suppression of false alarms [31]. Although using the Bayesian for the intrusion detection or intruder behavior prediction can be very appealing, there are some issues that one should be concerned about them. Because the accuracy of this method is dependant on certain assumptions that are typically based on the behavioral model of the target system, deviating from those assumptions will decrease its accuracy. Selecting an accurate model will lead to an inaccurate detection system. Therefore, selecting an accurate behavioral model is not an easy task as typical systems and/or networks are complex.

4.2.1. Bayesian belief network.

Bayesian belief networks (BBNs) are powerful tools for modeling causes and effects in a wide variety of domains. They are compact networks of probabilities that capture the probabilistic relationship between variables, as well as historical information about their relationships.

An important fact to realize about BBNs is that they are not dependant on knowing exact historical information or current evidence. A BBN is a model that represents the possible states of a given domain. A BBN also contains probabilistic relationships among some of the states of the domain. For example, when probabilities are entered into this BBN that represent real world weather and sprinkler usage, this belief network can be used to answer questions such as the following:

- If the lawn is wet, was it more likely to be caused by rain or by sprinkler?
- How likely is it that I will have to water my lawn on a cloudy day?

4.2.1.1. Model construction. The directed acyclic graph structure of BBNs contains nodes representing domain variables, and the arcs between the nodes represent probabilistic dependencies. During Bayesian network construction, a directed acyclic graph is built that encodes assertions of conditional independence. Because a Bayesian network for a dataset determines a joint probability distribution for the dataset, BBN is used to compute any probability of interest.

Once a BBN is constructed from prior knowledge or data, one can determine various probabilities of interest from the model. In our work, we have used a local score metric approach that aims to optimize the network structure on the basis of the quality of nodes as indicated by a given metric. The quality of the whole network is given by the sum of the individual nodes. A local search algorithm is used to compute the metrics for each node. In this paper, we use BBN with Tabu search (TS) as a feature selection algorithm to perform the classification to detect network intrusions efficiently.

4.3. Rule-based classifiers

In this section, we will focus on some very important and yet novel rule-based classification algorithms such as NNge and JRip, which are not yet explored by intrusion detection researchers to the best of our knowledge.

4.3.1. Non-nested generalized exemplars (NNge).

NNge is a novel algorithm that generalizes exemplars without nesting or overlap. NNge is an extension of Nge [32], which performs generalization by merging exemplars, forming hyperrectangles in feature space that represent conjunctive rules with internal disjunction. NNge forms a generalization each time a new example is added to the database, by joining it to its nearest neighbor of the same class.

Unlike Nge, it does not allow hyperrectangles to nest or overlap. This is prevented by testing each prospective new generalization to ensure that it does not cover any negative examples and by modifying any generalizations that are later found to do so. NNge adopts a heuristic that performs this post-processing in a uniform fashion. The summary of the NNge algorithm is described in the following text.

4.3.1.1. NNge algorithm description. NNge learns incrementally by first classifying and then generalizing each new example. It uses a modified Euclidean distance function that handles hyperrectangles, symbolic features, and exemplar and feature weights. Numeric feature values are normalized by dividing each value by the range of values observed. The class predicted is that of the single nearest neighbor. NNge uses dynamic feedback to adjust exemplar and feature weights after each new example is classified. When classifying an example, one or more hyperrectangles may be found that the new example is a member of, but which are of the wrong class. NNge prunes these so that the new example is no longer a member. Once classified, the new example is generalized by merging it with the nearest exemplar of the same class, which may be either a single example or a hyperrectangle. In the former case, NNge creates a new hyperrectangle, where as in the latter it grows the nearest neighbor to encompass the new example. Over generalization, caused by nesting or overlapping hyperrectangles, is not permitted. Before NNge generalizes a new example, it checks to see if there are any examples in the affected area of feature space that conflict with the proposed new hyperrectangle. If so, the generalization is aborted, and the example is stored verbatim. The more details about this algorithm can be found in [33].

4.3.2. Extended repeated incremental pruning (JRip).

JRip implements a propositional rule learner, "repeated incremental pruning to produce error reduction" (RIPPER), as proposed in [34]. JRip is a rule learner alike in principle to the commercial rule learner RIPPER. There are some characteristics of RIPPER algorithm that make it a very good choice for rule induction. The reasons are as follows:

- It can generate descriptive rules versus neural networks that are black box.
- It is a direct rule generator.
- It generates rules for classes with less distribution to more distribution, and the class with the most members is considered as default class.

JRip is classification inducer (requires a discretized classification value) that implements a propositional rule learner. It also uses the "RIPPER" to optimize the original version of incremental reduced error pruning (IREP). RIPPER rule learning algorithm is an extended version of learning algorithm IREP. It constructs a rule set in which all positive examples are covered, and its algorithm performs efficiently on large, noisy datasets. Before building a rule, the current set of training examples are partitioned into two subsets, a growing set (usually 2/3) and a pruning set (usually 1/3). The rule is constructed from examples in the growing set. The rule set begins with an empty rule set, and rules are added incrementally to the rule set until no negative examples are covered. After growing a rule from the growing set, condition is deleted from the rule to improve the performance of the rule set on the pruning examples. To prune a rule, RIPPER considers only a final sequence of conditions from the rule and selects the deletion that maximizes the function as given in Equation (2).

$$V(\text{Rule},\ \text{PrPos},\ \text{PrNeg}) = (p - n)/(p + n) \qquad (2)$$

where Rule is the set of rules, PrPos is the total number of examples in the considered cluster, PrNeg is the total number of examples in the cluster not considered, and p (or n) is the number of PrPos (or PrNeg) examples covered by Rule. Whenever no deletion improves the value of function w, learning process stops.

Furthermore, after the rule is added to the rule set, the total description length of the rule is computed. When the longest description length is more than 64 bits larger than the smallest one, learning also stops. All covered positive and negative examples are removed from growing and pruning set, and a new rule is constructed from the remaining examples. Because JRip considers classes with fewer members first, it generates more desirable rules, and it is more preferable than DTs.

4.4. Clustering approach

Clustering algorithms have gained much attention because they can help current intrusion detection systems in several aspects. Clustering aims to organize a collection of data items into clusters, such that items within a cluster are more "similar" to each other than they are to items in the other clusters. This notion of similarity can be expressed in very different ways, according to the purpose of the study, to domain-specific assumptions, and to prior knowledge of the problem. Clustering is usually performed when no information is available concerning the membership of the data items to predefined classes. For this reason, clustering is traditionally seen as part of unsupervised learning. An important advantage of using clustering or unsupervised learning to detect network attacks is because of its ability to detect attacks that were not seen before. The results of clustering can assist network security experts to label network traffic records as normal or intrusive. The amount of available network traffic audit data are usually large, making the expert-based labeling process of all records very tedious, time consuming, and expensive. Additionally, labeling a large number of network traffic records can lead to errors being incorporated during the process. Instead of evaluating each data instance one by one, the expert can simultaneously label all data instances in a cluster by observing the common characteristics of the cluster, possibly with very few errors, provided the clusters obtained are relatively pure. A completely pure

(100%) cluster is one that contains data instances only from one category (normal or attack).

4.4.1. Farthest first.

The FFT k-center algorithm is a fast, greedy algorithm that minimizes the maximum cluster radius [35]. This is also treated as an efficient algorithm that always returns the right answer. The pseudo-code for the FFT algorithm is shown in Figure 1.

Here, $\rho(x, T)$ is the distance from point (x) to the closest point in set T.

This builds a solution T one point at a time. It starts with any point and then iteratively adds in the point furthest from the ones chosen so far. The furthest point (x) from a set (S) is obtained from $\rho(x, T)$. FFT takes time $O(k|S|)$, which is fairly efficient and is always close to optimal solution, in the sense that if T is the solution returned by the FFT and T^* is the optimal solution, then $\cos t(T) \leq 2 \cos t(T^*)$.

5. HYBRID INTELLIGENT SYSTEM FOR DETECTING NETWORK INTRUSIONS

In this section, we investigate some novel hybrid intelligent systems by developing multiple classifiers in detecting network intrusions by using NSL-KDD dataset, a new variant of KDDCup 1999 dataset.

5.1. Combining decision tree with naïve Bayes (NBDT) and FFT (DTFF)

This paper combines NB with DT J48, which is called as hybrid NBDT, to build an efficient network intrusion detection model. This approach to the beat of our knowledge has not been used by any of the intrusion detection researchers using either KDDCup 1999 or NSL-KDD dataset. In this model, NB and decision tables can both be trained efficiently, and the same holds true for the combined model. Figure 2 shows the architecture of the proposed hybrid approach by combining DT with NB and then by combining DT with FFT clustering to obtain a hybrid DDFF.

The algorithm for learning the combined model (NBDT) proceeds in much the same way as the DTs alone. At each point in the search, it evaluates the merit associated with splitting the attributes into two disjoint subsets: one for the NB and the other for the DTs. In this, forward selection is used, where at each step, selected attributes are modeled by NB and the reminder by the DT, and all attributes are modeled by the DT initially. We use leave-one-out cross-validation to evaluate the quality of a split on the basis of the probability estimates generated by the combined model. In this, we aim to use accuracy as our performance measures in a two-class classification process in building a network intrusion detection system, as most of the anomaly detection schemes are concerned to obtain whether the particular instance belongs either normal or attack without discussing much insight about the attack types. The class probability estimates of the NB and DTs must be combined to generate overall class probability estimates. Further, we proposed to use FFT clustering in place of NB to take the advantage of classifying the rare attacks and then combined with DT to obtain a better model.

FARTHEST FIRST TRAVERSAL (FFT) ALGORITHMS

Pick any $z \in S$ and set $T = \{z\}$

While

$|T| < k$:

$z = \arg\max_{x \in S} \rho(x, T)$

$T = T \cup \{z\}$

Figure 1. Pseudo-code for farthest first traversal clustering.

5.2. Combining NNge with JRip and BBN

In the same way, we combined the rule-based classifiers such as the following: NNge and JRip to obtain a hybrid intelligent system (NNJR) for detecting network intrusions, which is shown in Figure 3. Here, JRip is used as a filter to reduce the amount of the data directly processed by the IDS, and then the obtained data are sent to NNge for proper classification. Similarly, we use FFT clustering as a filtering method and then use the NNge as a classifier to obtain a network intrusion detection system (FFNN). As the result, obtained data were not encouraging for NNge as a classifier

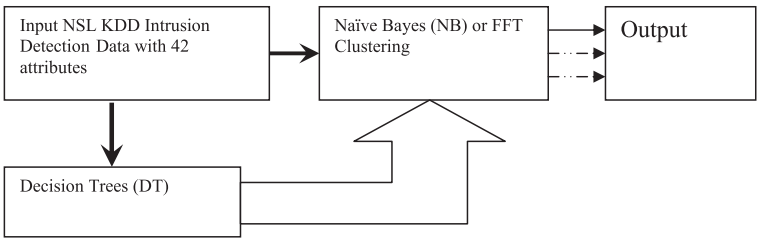

Figure 2. A hybrid intelligent approach by combining DT with NB and FFT.

combination strategies. Finally, we use NNge as a filter to combine with BBN and TS as a classifier to classify the instances in a more efficient way. This meta filtered classifier approach basically considered to have the advantage of decreasing storage requirements, reducing processing time and improving the detection rate.

6. EXPERIMENTAL RESULTS AND DISCUSSIONS

The experiments were conducted with NSL-KDD dataset, a variant of KDDCup 1999 benchmark intrusion detection dataset. We have performed two-class classification (i.e., either normal or anomaly) to build our proposed hybrid intelligent system to detect network intrusions. All experiments were conducted on a Pentium-4 IBM PC with 2.8 GHz CPU, 40 GB HDD with 512 MB RAM. We have used Weka 3.7 [36] with all default values for our proposed methodologies.

To solve the issues mentioned earlier, we performed our experiments by using two new hybrid approaches for detecting network intrusions on NSL-KDD dataset that contains 25,192 training and 22,544 separate testing instances with 41 input attributes and a class label either normal or anomaly as output attribute. All attributes are same to that of KDDCup 1999 dataset. To test and compare the effectiveness of the proposed

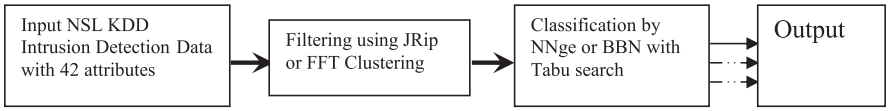

Figure 3. Hybrid approach by combining rule-based classifiers with clustering and BBN.

Table I. Comparison of hybrid algorithms with NSL-KDD dataset.

Algorithms using separate testing dataset with NSL-KDD dataset	Normal		Anomaly		Build time in seconds	RMSE
	DR (%)	FPR (%)	DR (%)	FPR (%)		
Naïve Bayes	92.78	13.82	64.3	26.22	3.52	0.4816
Decision trees (J48)	97.17	5.1	68.86	23.29	26.89	0.413
NNge	91.25	13.2	68.47	25.12	42.92	0.466
JRip	97.15	5.24	68.42	23.48	162.84	0.4373
NBDT	97.17	5.1	68.86	2.83	29.11	0.413
NNJR	91.25	13.2	68.47	25.13	242.89	0.446
AdaBoost + NBDT	97.33	5.44	66.52	24.15	553.34	0.4399
DTFF	97.16	4.64	71.03	22.4	37.23	0.418
FFNN	91.25	13.18	68.5	25.11	48.94	0.466
BBN + Tabu search + NNge	98.31	17.87	69.12	23.87	92.24	0.5

Table II. Performance evaluation using KDDCup 1999 dataset.

Algorithms using separate testing dataset with KDDCup'99 dataset	Normal		Anomaly		Build time in seconds	RMSE
	DR (%)	FPR (%)	DR (%)	FPR (%)		
Naïve Bayes	92.8	35.7	64.3	7.2	4.38	0.4816
Decision trees (J48)	97.2	31.1	68.9	2.8	33.59	0.413
NNge	91.2	31.5	68.5	8.8	45.84	0.466
JRip	97.1	31.6	68.4	2.9	172.8	0.4373
NBDT	97.2	31.1	68.9	2.8	38.19	0.413
NNJR	91.2	31.5	68.5	8.8	215.27	0.466
AdaBoost + NBDT	97.3	33.1	66.9	2.7	208.5	0.4338
DTFF	97.2	31.2	68.8	2.8	39.55	0.4329
FFNN	91.2	31.5	68.5	8.8	52.36	0.4661
BBN + Tabu search + NNge	97.3	42.7	57.3	2.7	86.47	0.5

methodologies, we have used separate testing dataset for designing the intrusion detection model. The results obtained using these are provided in Table I for individual and multiple classifier systems.

On the basis of the critiques provided in [9,10], we test our model by using separate NSL-KDD testing dataset with 22,544 instances, which are different than the training dataset used, which is provided in Table I. With this approach, the detection rate for NBDT and NNJR restricted to only 68.86% and 68.47%, respectively. The result shows that meta filtered classifier using NNge cannot enhance the detection rate as we use rule-based classifier for both filtering and classification purpose. It can also be noted from Table I that the ensembling NBDT using AdaBoost using separate testing dataset does not enhance the performance of the proposed hybrid intelligent system. Next, for further exploration on choosing a perfect combination method, we tried to use BBN with TS for classifying the instances obtained after filtering is carried out using NNge classifier. The result shows that BBN with TS using NNge as filter could able to detect 98.31% normal instances in comparison with 91.25% for single NNge and 69.12% for anomaly detection with 68.43% for NNge only. Moving ahead with the proposal to use combination of classifiers for enhancing the overall performance of NIDS, we use clustering techniques such as FFT to combine with a classifier technique for the purpose. As is evident from Table I, the detection rate for anomaly behavior in case of DTFF is 2.2% better than the NBDT approaches, and for FFNN, it is 0.3% better than NNJR.

Other performance measures such as model building time with 37.23 s and root mean square error of 0.418 for building NIDS using hybrid DTFF is found almost same with NBDT.

The results obtained after using KDDCup 1999 dataset is also provided in Table II so as to compare with results obtained in Table I with NSL-KDD dataset. It is also evident from the comparison that DTFF and BBN + TS + NNge provide better results with NSL-KDD dataset than KDDCup 1999 in terms of detection rate, with acceptable false positive rate.

All these analysis laid a foundation on investigating many more methodologies to find a suitable model to detect network intrusion efficiently with 100% detection rate and 0% false positive rate. Further, it is very difficult on our part to compare the proposed methodologies with others because of unavailability of the research works using the NSL-KDD dataset to detect network intrusions.

7. CONCLUSION AND FUTURE SCOPE

In this paper, we tried to develop some hybrid approaches by combining classifiers such as NBDT, NNJR, and BBN + NNge and then by combination of clustering with classifier such as DTFF and FFNN to detect network intrusion efficiently. We used NSL-KDD dataset, a new variant of KDDCup 1999 dataset, for our proposed approaches by using separate testing dataset to build the network intrusion detection model. While analyzing the results, it is understood that the combination of clustering with classification techniques provides better result than classifier combination strategies. Although the detection accuracy for anomaly is around 71%, still this can be considered to be good in this area of research, so to increase the detection rate to a greater extent further while maintaining a low positive rate, more data-mining approaches are to be explored in the future.

REFERENCES

1. Adetunmbi AO, Zhiwei S, Zhongzhi S, Adewale OS. Network anomalous intrusion detection using fuzzy-Bayes. In *IFIP International federation for information processing*, 3rd edn, Vol. 228. Intelligent Information Processing, Shi Z, Shi-mohara K, Feng D (eds). Springer: Berlin, 2006; 525–530.
2. Ho (George) SY. Intrusion detection-systems for today and tomorrow. SANS institute, 2001. Accessed online on: April 10, 2010. Available at: http://www.sans.org/reading_room/whitepapers/detection/intrusion-detection-systems-today-tomorrow_341
3. Byunghae-cha KP, Jaittyun S. Neural network techniques for host anomaly intrusion detection using fixed pattern transformation. In *ICCSA 2005*, Lecture Notes in Computer Science, Vol. 3481, 2005; 254–263.
4. Biswanath M, Todd LH, Karl NL. Network intrusion detection. *IEEE Network* 1994; **8**(3): 26–41.
5. Xu X, Wang XN. Adaptive network intrusion detection method based on PCA and SVM. In *BNAI (ADMA-2005)*. Lecture Notes in Artificial Intelligence, Vol. 3584, 2005; 696–703.
6. Shon T, Soe J, Moon J. SVM approach with a genetic algorithm for network intrusion detection. In *Proceedings of 20th International Symposium on Computer and Information Sciences* (ISCIS-2005). Springer: Berlin, 2005; 224–233.
7. Ghosh KA, Schwartzbard A. Study in using neural networks for anomaly and misuse detection. In *Proceedings of the 8th USENIX Security Symposium*, Washington DC, August 1999; 131–142.
8. KDDCup 1999 Data. Available at: http://www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
9. McHugh J. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluation as performed by Lincoln laboratory. *ACM Transaction on Information and System Security* 2000; **3**(4): 262–294.

10. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of KDDCup 99 dataset. In *Proceedings of 2009 IEEE Symposium on Computational Intelligence in Security and Defence Applications* (CISDA-2009), IEEE Press: USA, 2009.
11. Fisch D, Hofmann A, Sick B. On the versatility of radial basis function neural networks: a case study in the field of intrusion detection. *Information Sciences* 2010; **180**: 24121–2439.
12. Zanial A, Maarof MA, Shamsuddin SM. Ensemble classifiers for network intrusion detection system. *Journal of Information Assurance and Security* 2009; **4**: 217–225.
13. Mukkamala S, Sung AH, Abraham A. Intrusion detection using an ensemble of intelligent paradigms. *Journal of Network and Computer Applications* 2005; **28**: 167–182.
14. Panda M, Patra MR. *A semi-naïve Bayesian approach for network intrusion detection system*. Lecture Notes in Computer Science, Vol. 5863, Springer: Berlin, 2009; 614–621.
15. Panda M, Patra MR. A hybrid clustering approach for network intrusion detection using COBWEB and FFT. *Journal of Intelligent Systems* 2009; **18**(3): 229–245.
16. Tran TP, Cao L, Tran D, Nguyen CD. Novel intrusion detection using probabilistic neural network and adaptive boosting. *International Journal of Computer Science and Information Security* 2009; **6**(1): 83–91.
17. Farid D, Harbi M, Rahman MZ. Combining naïve Bayes and decision trees for adaptive intrusion detection. *International Journal of Network Security and its Applications* 2010; **2**(2): 12–24.
18. Panda M, Patra MR. Ensembling rule based classifiers for detecting network intrusions. In *Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing ARTCom-2009*. IEEE Press: USA, 2009; 19–22.
19. Powers ST, He J. A hybrid artificial immune system and self organising map for network intrusion detection. *Information Sciences* 2008; **178**: 3024–3042.
20. Shon T, Moon J. A hybrid machine learning approach to network anomaly detection. *Information Sciences* 2007; **177**: 3799–3821.
21. Mahoney M, Chan P. *An analysis of the 1999 DARPA/ Lincoln laboratory evaluation data for network anomaly detection*. LNCS, 2003; 220–238.
22. Khirsagar VP, Patil DR. Applications of variant of AdaBoost based machine learning algorithm in network intrusion detection. *International Journal of Computer Science and Security* 2010; **4**(2): 1–6.
23. Kou G, Peng Y, Chen Z, Shi Y. Multiple criteria mathematical programming for multiclass classification and application in network intrusion detection. *Information Sciences* 2009; **179**:371–381.
24. STEAL (Security Technology Education and Analysis Laboratory), Nebraska University Consortium on Information Assurance (NUCIA), 2005. Available at: http://nucia.ist.unomaha.edu/steal/labs.php
25. Stofo SJ, Fan W, Lee W, Prodromidis A, Chan PK. Cost based modelling and evaluation for data mining with application to fraud and intrusion detection, JAM project, 1999; 1–39.
26. Witten IH, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques* (2nd edn). Morgan Kaufmann, Elsevier: US, 2005.
27. Hackermann D. A tutorial on learning with Bayesian networks, Microsoft research. *Technical Report*, MSR-TR-95-06, 1995.
28. Kragel C, Toth T, Kirda E. Service specific anomaly detection for network intrusion detection. In *ACM Symposium on Applied Computing*. ACM Press: Madrid, Spain, 2002; 201–208.
29. Valdes A, Skinner K. Adaptive model based monitoring for cyber attack detection. In *Recent Advances in Intrusion Detection*, LNCS, Vol. 1907, Springer Verlag: Berlin, 2000; 80–92.
30. Ye N, Xu M, Emran SM. Probabilistic networks with undirected links for anomaly detection. In *IEEE Systems, Man and Cybernatics Information Assurance and Security Workshop*, West Point, NY, 2000.
31. Panda M, Patra MR. Mining knowledge from network intrusion data using data mining techniques. In *Knowledge Mining using Intelligent Agents*, Dehuri SN, Chao S-B (eds). Chapter. 6, Imperial College Press: London, 2010; 161–200.
32. Salzberg S. A nearest hyperrectangle learning method. *Machine Learning* **6**: 277–309.
33. Roy S. *Nearest neighbour with generalization*. University of Canterbury: Christchurch, NZ, 2002.
34. Cohen WW. Fast effective rule induction. In *12th International Conference on Machine Learning*. 1995; 115–123.
35. Willium A. Clustering algorithm for categorical data, 2006.
36. Weka: Waikato environment for knowledge analysis, version 3.7.1. Available at: http://www.cs.waikato.ac.nz/ml/weka/, February 20, 2010.