# Parallel Misuse and Anomaly Detection Model

Radhika Goel, Anjali Sardana, and Ramesh C. Joshi
*(Corresponding author: Radhika Goel)*
Department of Electronics and Computer Science Engineering, Indian Institute of Technology Roorkee
Roorkee (Uttranchal), 247667, India
(Email:{radhikaiitr, dr.anjalisardana}@gmail.com, rcjosfec@iitr.ernet.in)
*(Received Aug. 16, 2010; revised and accepted Feb. 11, 2011)*

## Abstract

In this paper a novel hybrid model is being proposed for misuse and anomaly detection. C4.5 based binary decision trees are used for misuse and CBA (Classification Based Association) based classifier is used for anomaly detection. Firstly, the C4.5 based decision tree separates the network traffic into normal and attack categories. The normal traffic is sent to anomaly detector and parallel attacks are sent to a decision trees based classifier for labelling with specific attack type. The CBA based anomaly detection is a single level classifier where as the decision trees based misuse detector is a sequential multi-level classifier which labels one attack at a time in a step by step manner. The model is trained and tested on two disjoint datasets provided in the KDD Cup 99. Results show that 99.995% misuse detection rate with an anomaly detection rate of 99.298% is achievable. The overall attack detection rate is 99.911% and false alarm ratio of the integrated model is 3.229%. To overcome the deficiencies in KDD 99 dataset, a new improved dataset is also proposed. The overall accuracy of integrated model trained on new dataset is 97.495% compared to 97.24% of the old dataset.

*Keywords: CBA based classifier, data mining, decision tree C4.5 algorithm, KDD dataset, misuse and anomaly detection, network security*

## 1 Introduction

With the enormous growth of computer networks, network security is becoming more and more challenging. In addition to intrusion protection techniques (like user authentication, user authorization, encryption and defensive programming) intrusion detection and network forensics are also needed for complete security of network.

The effectiveness of an Intrusion Detection System is measured using its likelihood of giving a signal upon an intrusion i.e. attack detection rate and the ratio of false alarms in them [3]. On the other hand, network forensics is about offline investigation on captured data to discover the source of security attacks [6]. For this kind of offline and online traffic analysis, the detection of an attack and its correct categorization is crucial.

Recently, a great interest in application of data mining techniques for attack detection and identification purposes [4] has been seen. The problem of attack detection can be reduced to a data mining task of classifying data. Briefly, given a set of data points belonging to different classes (normal activity and attacks of different types) one has to separate them as accurately as possible by means of a model.

There are two different approaches used to identify attacks- Misuse detection where attacks are detected using their known signatures. In Anomaly Detection method, firstly a normal system behavior profile is created and any deviation from that defined profile as marked as anomaly.

In this paper we propose a novel parallel classification model for misuse and anomaly attack detection using data mining techniques.

## 2 Background

Many key researches have been done in network intrusion detection area. Most of these works have used KDD cup 99 dataset for evaluation of their attack detection methods. Some researches that have been done on KDD dataset are highlighted in this section.

Nguyen and Choi [13] compared different algorithms on the basis of their percentage accuracies of individual attack type detection and on the basis of overall accuracies (AA). Algorithms' classification times (TT in sec) were also compared to find their real time usage. Compared algorithms included BayesNet AA(90.62) TT(6.28), NaiveBayes AA(78.32) TT(5.57), C4.5 AA(92.06) TT(15.85), NBTree AA(92.28) TT(295.88), Decision Table AA(91.66) TT(66.24) and few others. Even though the overall accuracy of single classifiers, mainly C4.5, was quite good but none of them was able to detect all four attacks efficiently.

Tavallaee et al. [17] also conducted similar experiments

on KDD dataset. In their results also C4.5 gave the highest overall accuracy of 93.82% among the considered classifiers like NaiveByes (81.66%), NB Tree (93.51%), Random Forests (92.79%), Random Tree (92.53%), Multilayer Perceptron (92.26%), and Svm (65.01%).

Depren et al. [7] proposed an IDS architecture utilizing Self-Organizing Map (SOM) structure for anomaly and C4.5 for misuse detection. A rule based Decision Support System (DSS) was also developed for interpreting the results of both anomaly and misuse detection modules. They obtained a detection rate of 98.96% for anomaly detection and 99.61% for the misuse detection modules on the KDD 99 Data Set.

Another model for IDS, proposed by Pan et al. [14], used neural network and C4.5 for attack detection. Their model achieved the average detection rate of 93.28% and false positive rate of 0.2% on KDD Cup 99 dataset. Kandeeban and Rengan [9] used a combination of genetic algorithm and neural networks for intrusion detection on KDD 99 dataset. They achieved a detection rate as close to 95 when the false alarm rate is 1.9% to 2% and a detection arte of 70% as the false alarm rate is brought down to 1%.

Hon and Li [8] used the hybrid of C4.5 and Chi-Square for selecting most useful features for attack detection. They also evaluated there model using KDD Cup 99 dataset. Another paper by Peddabachigari et al. [15] used C4.5 Decision trees (DT) and support vector machines (SVM) hybrid model for attack detection.

As discussed above, many researchers have conducted extensive performance comparison of various popular classification algorithms. The decision tree based algorithms like C4.5 have seemed to give the best performance so far as single level classifiers.

C4.5 [18] is an open-source widely used classifier for attack detection in network traffic. It is a freely available data mining tool that is descended from an earlier system called ID3 and is followed in turn by See5/C5.0.

C5.0/See5 [5] is a commercial and closed-source product, although its source code is available free of charge. C5.0/See5 gives similar results to C4.5 with considerably smaller decision trees. Moreover, the boosting feature of C5.0 improves the trees and gives them more accuracy.

However, it has also been seen that a single level classifier cannot be applied to address all different attack categories efficiently. Moreover, decision tree based classifiers can detect only what they have learnt. They fail to detect anomalies efficiently. Numerous hybrid models have been proposed till now (as discussed above) that use algorithms like SVM, SOM, neural networks, as anomaly detectors in combination with C4.5 based misuse detection part.

Efficiently detecting outliers or anomalies is an important problem in many areas of science, and information technology. Asfa et al. [2] used rules based classifier - CBA (Classification based Association) for anomaly detection in Pervasive Medical Systems. Ma et al. [20] also used rules based classifier - CBA for intrusion detection in Ad hoc networks.

In this paper, we have proposed a novel hybrid model using rules based classifier for anomaly detection and decision trees in a sequential multi-level model for misuse detection. In Section 3, the multi-level hybrid model is discussed in detail.

To evaluate our hybrid model on the base line of other hybrid models built using C4.5 decision tree algorithm, we have also used C4.5 for misuse detection in our multi-level model. However, because of C5.0's advantages over C4.5, C5.0 can also be equally used in place of C4.5 in the proposed model. For anomaly detection, we have used CBA (Classification based Association) as rules based classifier. The C4.5 and CBA classification algorithms are discussed in Section 4 of this paper.

The KDD'99 dataset, the benchmark dataset on which all above discussed models are evaluated, is the most widely used data set and is among the few datasets that can easily be shared with other researchers, allowing all kinds of techniques to be easily compared in the same baseline. For these reasons, we have also used the KDD CUP 99 dataset, prepared by Stolfo et al. [10, 11], to validate the efficiency of our proposed model. In Section 5, the two disjoint datasets used for training and testing are described. Experimental results are discussed in Section 6.

However, it is important to note that the KDD '99 dataset suffers from some potential problems [16]. Therefore, taking into account the issues associated with the credibility of the KDD dataset, we have tried to identify those key problems in the dataset in Section VII and have proposed a new improved version of the KDD dataset. Then, we have evaluated our model on the newly improved version.

Section 8 draws the conclusions and outlines the possible future work.

# 3 Multi-Level Parallel Classification Model

To build an efficient attack detection model, we need to take into account two kinds of attack patterns. The one kind of traffic has known attack patterns. The training data has patterns similar to them and a classifier can be made to learn those signatures. This is called Misuse or Signature based Detection. Such signature- based classifiers have high detection rates for known attack patterns but fail drastically to detect patterns that are new to them. These attacks that fall into some specific attack categories but whose attacking patterns are new to the model are called Anomalies. Recently more and more research is going on to build models that detect such anomalies with high detection rate.

We are proposing a new approach that categorizes known attack traffic into different attack types in a sequential manner and in parallel separates out anomalies from normal traffic.

### 3.1 Sequential Misuse Detection Model

A hierarchical sequential model with binary decision tree classifiers at each level is proposed. The differential approach separates out one attack at a time. This technique defines the unique features of one attack and at the same time brings about the general characteristics of rest of the other attacks which differentiate the rest from that attack.

The distribution of different attacks in training dataset is usually uneven. The instances of some attacks are often less than others. In this model, sequence maintained at different levels is so as to make an unbiased classifier by combining the less frequent records together.

Moreover, the sequential nature of the proposed multi-level architecture needs binary classification at each level. Decision trees generated using this sequential technique is small and more interpretable than the one big tree. Small trees can reside in the memory all at a time and involve less input-output operations leading to faster classification needed for online purposes.

Figure 1 shows the prototype of sequential model. At each level one attack class is being separated from the rest of the data.

In the proposed architecture, classic C4.5 (J48) [18] is being used as the decision tree algorithm. We have used Weka's J48 algorithm, the Weka's version of C4.5, with default parameters [19]. *J48* is an optimized implementation of C4.5.

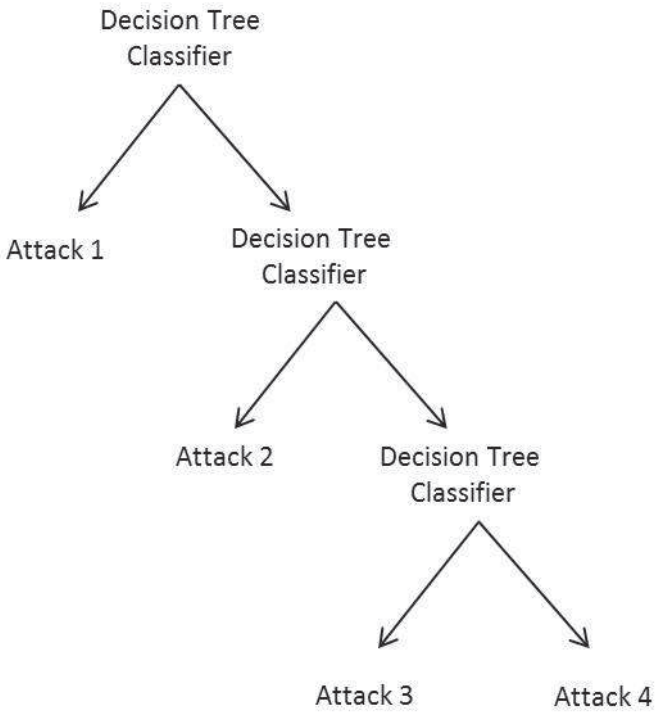


Figure 1: Sequential multi- level misuse detection model.

### 3.2 Anomaly Detection Model

For anomaly detection, a rules-based classifier is used. The rules for normal profile are defined and test traffic is tested against them. If data instance doesn't satisfy any of the normal profile rules, then it is considered as anomaly (refer to Figure 2).

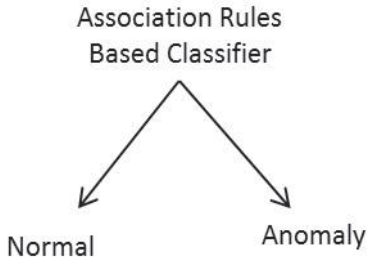


Figure 2: Rule based anomaly detection model.

Our approach uses Association classification, where association rules are generated and analyzed for use in classification. These classifiers search for strong associations between frequent patterns (conjunctions of attribute-value pairs) and class labels. Because association rules explore highly confident associations among multiple attributes, this approach may overcome some constraints introduced by decision-tree induction, which considers only one attribute at a time.

Association rules for profiling are generated using CBA (Classification Based Association) algorithm [12]. CBA generates rules using real traffic i.e. labelled training data consisting of both normal and attack traffic patterns. The advantage of taking both classes in training data is that only strong rules that differentiate normal and attack traffic will be generated based upon higher support and confidence value. From these rules, rules that define normal traffic class are taken out to make normal profile and test traffic is tested against them for deviation.

### 3.3 Integrated Misuse and Anomaly Detection Model

The Misuse and Anomaly detection Model is integrated by a Decision Tree based classifier at first level. It separates out attack traffic (of known signature) from normal traffic (containing both anomalies and normal traffic). The association rules based Anomaly detector and decision tree based Misuse Detector are placed at second level. The suspected data (outputted as normal by first level decision tree) is sent to the anomaly detector and the known attack data (outputted as attack by first level decision tree) is sent to the Misuse detector to be classified into different categories (Refer to Figure 3).

A score or alarm rating can be assigned to attacks (known attacks and anomalies) detected to determine how known the attack is compared to anomalous traffic.

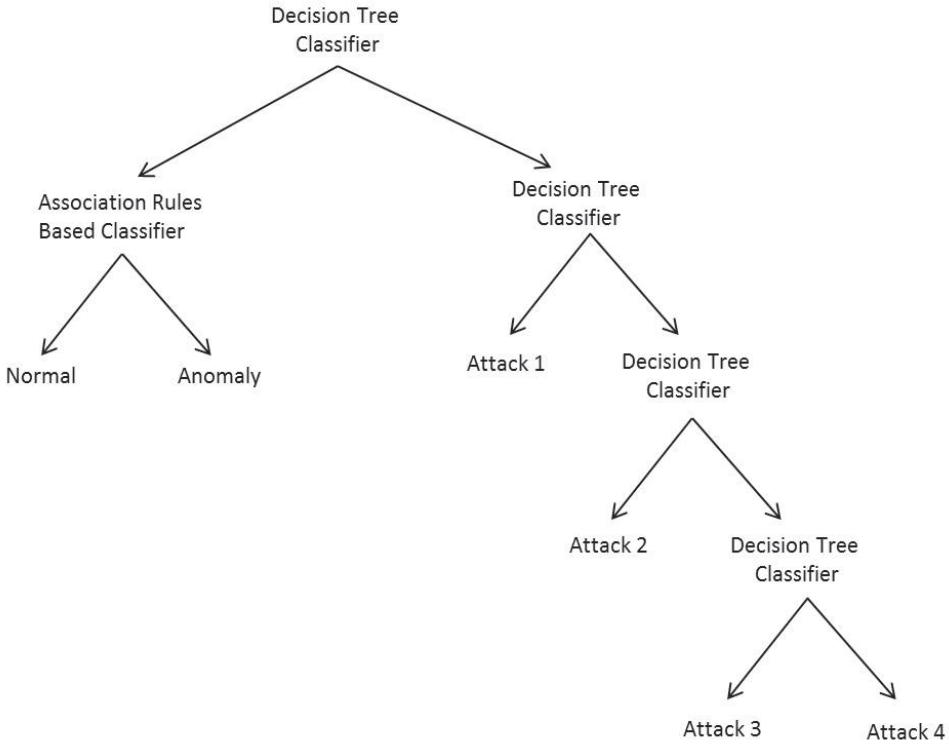


Figure 3: Misuse and anomaly detection model.

The placement of decision tree based classifier (a very strong classifier) at first level ensures that traffic with only known attack signatures are placed in attack category. This class would have very little of normal or anomalous traffic i.e. the false positive rate is very low. These attacks are of known nature and are given high score or high alarm rating.

As only normal profile rules are used for testing at anomaly detector, its false positive rating is usually high and so the attacks identified by it are given low alarm rating.

Hence, this parallel Misuse and Anomaly Detection model can be applied to any dataset with different attack types (with known and new attack patterns).

# 4 Classification Algorithms

## 4.1 Standard Decision Tree Algorithm: C4.5 (J48)

Decision tree based methods are widely used in data mining and decision support applications. C4.5 algorithm developed by Quinlan [18] is the most popular decision tree classifier. It uses the concept of entropy, to measures the impurity of data items.

The value of entropy is small when the class distribution is uneven and is high when the class distribution is more even.

Information gain is calculated that measures the decrease of the weighted average impurity (entropy) of the attributes compared with the impurity of the complete set of data items. Therefore, the attributes with the largest information gain are considered as the most useful for classifying the data items.

Let $RF(C_j, S)$ denote the relative frequency of cases in $S$ that belong to class $C_j$. The information content of a message that identifies the class of a case in $S$ is then

$$l(S) \quad = \quad \sum_{j=1}^{x} RF(C_i, S) \log(RF(C_i, S)).$$

After $S$ is partitioned into subsets $S_1$, $S_2$, ..., $S_t$ by a test $B$, the information gain is

$$G(S, B) \quad = \quad l(S) - \sum_{i=1}^{t} \frac{|S_i|}{|S|} I(S_i).$$

Then to control the number of partitions i.e. $S_1$, $S_2$ etc $P(S, B)$ is calculated :

$$P(S, B) \quad = \quad -\sum_{i=1}^{t} \frac{|S_i|}{|S|} \log(\frac{|S_i|}{|S|}).$$

The test $B$ that maximizes $G(S, B)$ / $P(S, B)$ is then chosen as current partitioning attribute.

### 4.2 Standard CBA (Classification Based Association)

Classification based on association [12], also called associative classification, is the application of association rules to classification problems. It generates class association rules (CARs).

Classification association rules (CARs) are association rules with the target class on the right hand side of the rules. A CAR is an implication of the form:

$$X \rightarrow y, \quad where \quad X \subseteq I, \quad and \quad y \in Y.$$

$X$ is a set of features. $I$ is the set of all features. $y$ is the target class. $Y$ is the set of all classes.

CBA also provides strength measurements for the CARs:

**Support.** The rule holds with support sup, if sup% of cases contain $X$.

**Confidence.** The rule holds with confidence conf, if conf% of cases that contain $X$ also contain $y$.

The algorithm used for rule generation in CBA is similar to the Apriori algorithm [1] using generate and test approach. Firstly, size-$k$ patterns are generated (starting from size one). These are called candidate patterns. Then using apriori approach, candidate patterns satisfying minimum support are selected as frequent patterns. Using size-$k$ frequent pattern, size $k+1$ candidate patterns are generated and tested for minimum support value. This process continues till a max limit on rule size is reached or no frequent patterns exist. Finally from frequent patterns, rules are generated using confidence value. For details of Apriori algorithm and CBA rule generation algorithm refer to papers by $R$. Agrawal and $R$. Shrikant [1], and Bing Liu et al. [12] respectively.

After rule generation, CBA uses a heuristic method to order the rules in decreasing precedence based on their confidence and support values. If a set of rules has the same antecedent then the rule with the highest confidence is selected to represent the set. If the confidence of the rules that apply is the same, the rule with highest support will be picked. Again if the support is also equal, CBA will classify the case according to the rule which is generated earlier than the others. In this way an ordered list of rules is created. In this way an ordered list of rules is created.

When a new tuple is given for classification, the class associated with first rule satisfying the tuple is used for labelling. The classifier also contains a default rule, having lowest precedence. If a tuple doesn't satisfy any rule then it is assigned the label of default class.

## 5 Classification Model on Darpa KDD 99 Dataset

### 5.1 KDD Cup 99 Dataset

The KDD Cup 99 Dataset has disjoint training and testing datasets [10, 11].

The 10% KDD dataset is used for training. The training dataset has approximately 4,900,000 single connection vectors each of which contains 41 features. Each vector is labelled as either normal or an attack, with exactly one specific attack type. A smaller version 10% training dataset is also provided for memory constrained machine learning methods. The training dataset has 19.86% normal and 80.14% attack connections.

In the testing dataset, there are 19.48% normal, 74.50% old attack and 6.02% new attack connections which have not been shown in training set. Moreover, the probability distribution of different connections is also not same in test and training datasets which make the task more realistic.

The simulated attacks fall in one of the following four categories:

1) Denial of Service attack (Dos): In these attacks attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

2) Probe attacks (Probe): This is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

3) User to Root Attack (U2R): is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

4) Remote to Local Attack (R2L): occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

KDD'99 features can be classified into three groups:

1) Basic Features: This category encapsulates all the attributes that can be extracted from a TCP/IP connection.

2) Traffic Features: This category includes features that are computed with respect to a window interval and is divided into two groups: Time-based traffic features, stats about the connections in the past two seconds and Host-based traffic features, constructed using a window of 100 connections to the same host instead of a time window.

3) Content Features: Using the domain knowledge, content features are derived that look for suspicious behavior in the data portions, such as the number of failed login attempts.

### 5.2 Application of Misuse Detection Model on KDD Dataset

The 41 features in the KDD dataset are chosen according to the characteristic of different attack types. Some features help to identify a specific class of attacks and some are common to different classes.

The proposed sequential approach as being applied on KDD Dataset separating one attack at a time is explained below:

1) First Stage: The training data has only two classes - attack class and normal class. First classifier is trained on this data. The generated classifier has characteristics that are common to all different attacks and characteristics that distinguish attacks from a normal traffic.

2) Second Stage: The Dos attack uses more of time-based traffic features unlike Probe, U2R and R2L attacks that use host-based features and content features, respectively. Moreover, the Dos attack instances in the training data are more than the combined number of Probe, U2R and R2L attacks instances due to the common nature of Dos attacks. Hence to make an unbiased classifier, Dos attacks are separated from other attacks at second level.

3) Third Stage: Host- based traffic features are needed to identify some of the Probe attacks because they involve scanning of hosts (or ports) using a much larger time interval than two seconds. R2L and U2R attacks are generally defined using content features. Therefore, Probe attacks are taken separately and U2R and R2L attacks are put in one separate category.

4) Fourth Stage: A last tree does final classification between U2R and R2L attacks.

Figure 4 shows the sequential model being applied on KDD dataset.

### 5.3 Application of Anomaly Detection Model on KDD Dataset

The new attack patterns in KDD testing dataset fall into the same four categories - (DoS, Probe, U2R and R2L). CBA classifier is used to detect theses new attack patterns of as Anomaly.

The suspected traffic (outputted as normal by first level C4.5 based classifier) is taken as input by CBA (refer to Figure 5) for testing.

It takes out anomalies from suspected traffic and allows normal traffic to pass through it.

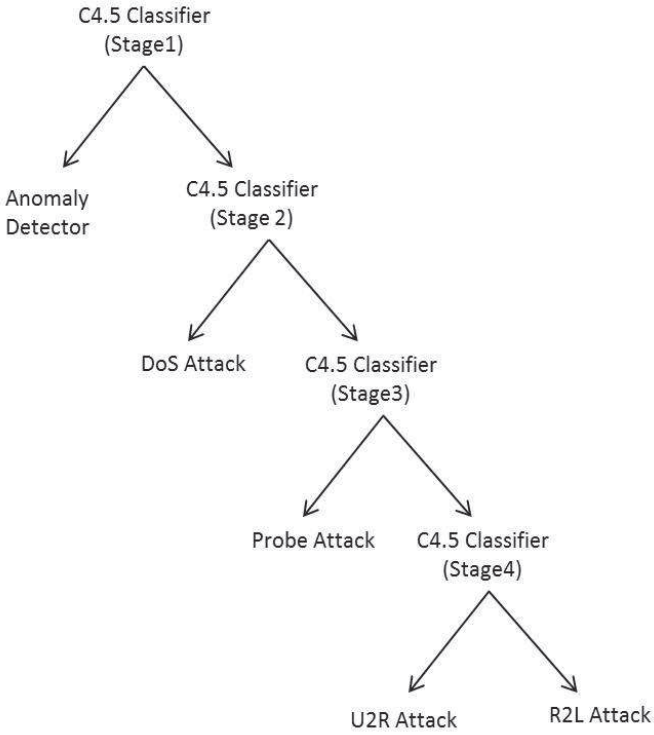


Figure 4: Misuse detection model for KDD dataset.

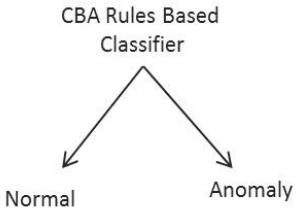


Figure 5: Anomaly detection model for KDD dataset.

## 6 Experimental Results on 10% Training Dataset and Test Dataset

### 6.1 Evaluation Metric

Metrics used to analyse the performance of the proposed model are Classification Accuracy, Precision and Recall. The basic data structure used for evaluation is Confusion Matrix.

Confusion Matrix: Table 1 shows a confusion Matrix. In the context of intrusion detection system true positive (TP) means correctly identified attack traffic and true negative (TN) means correctly identified normal traffic. False positive (FP) is the normal traffic being misclassified as an attack and false negatives (FN) are the attack instances being misclassified as normal.

Table 1: Confusion matrix

| Actual Class | Classified as | |
|---|---|---|
| | **Normal** | **Attack** |
| Normal | True Negative | False Positive |
| Attack | False Negative | True Positive |

Table 2: Training result of level 1 classifier

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | **Normal** | **Attack** | |
| Normal | 97239 | 39 | 97278 |
| Attack | 93 | 396650 | 396743 |
| Total | 97332 | 396689 | 494021 |

| | |
|---|---|
| Correctly Classified Instances | 99.973 % |
| Incorrectly Classified Instances | 0.0267 % |
| Attack Detection rate | 99.976% |
| False Alarm Ratio | 0.0098% |

**Classification Accuracy.** It is the percentage of correctly classified instances: $[(TP+TN)/(TP+TN+FP+FN)] * 100$.

**Precision.** Precision measures the proportion of predicted positives/negatives which are actually positive/negative. It can be defined as True Alarm Ratio: $TP/(TP+FP)$ or False Alarm Ratio: $FP/(FP+TP)$.

**Recall.** It is the proportion of actual positives/negatives which are predicted positive/negative. The Attack Detection rate or True Alarm rate is $TP/(TP+FN)$. False positive rate or False Alarm rate is: $FP/(FP+TN)$ and False negative rate is $FN/(FN+TP)$.

It is required from good IDS that Attack Detection rate ($TP$ / ($TP + FN$)) is as high as possible and the False alarm ratio ($FP$ / ($FP$ + $TP$)) is as low as possible. That is DS should generate alarm on most of the attacks and the ratio of false alarms in them should be least.

## 6.2 Misuse Detection Results

The KDD 10% training dataset is used to train the first classifier (refer to Table 2). All the traffic in this training dataset is labelled as attack or normal. Then this trained classifier is used to label the traffic in KDD Test dataset. Table 3 shows the testing results of Level 1 classifier. Out of 250436 instances of attack (anomalies + known attacks), 227752 instances are detected making an attack detection rate of 90.942%. The false alarm ratio is as low as 0.149%.

Out of 220525 attacks of known type from testing dataset, 219847 are detected by Level 1 classifier, making a Misuse detection rate of 99.683% (as shown in Table 4).

Table 3: Testing dataset of level 1 classifier

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | **Normal** | **Attack** | |
| Normal | 60253 | 340 | 60593 |
| Attack | 22684 | 227752 | 250436 |
| Total | 82937 | 228092 | 311029 |

| | |
|---|---|
| Classified Instances | 92.5975 % |
| Incorrectly Classified Instances | 7.4025 % |
| Attack Detection rate | 90.942% |
| False Alarm Ratio | 0.1490% |

Table 4: To-be tested audio files

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | **Normal** | **Attack** | |
| Known attack | 678 | **219847** | 220525 |
| Anomaly | **22006** | 7905 | 29911 |
| Total | 22684 | 2277752 | 250436 |

| | |
|---|---|
| Attack Detection rate | 90.942% |
| Misuse Detection rate | 99.683% |
| Anomaly Detection rate | 26.428% |

It also shows that anomalous data forms 97.011% of false negatives.

The second classifier (Level 2 classifier) is trained on only attack data with two classes - Dos and Other Attacks (refer to Table 5). The attacks of Level 1 classifier's Test results make the testing data for the second classifier. Table 6 shows the testing results of Level 2 classifier. Out of 223319 instances of Dos attack in the test dataset, 222524 instances are detected making a Dos attack detection rate of 99.644%.

Table 5: Confusion matrix of testing dataset for level 2 classifier

| Actual Class | Classified as | | | Total |
|---|---|---|---|---|
| | **Normal** | **Other Attacks** | **Dos Attack** | |
| Normal | 0 | 0 | 0 | 0 |
| Other Attacks | 0 | 35256 | 29 | 35285 |
| Dos Attack | 0 | 13 | 391445 | 391458 |
| Total | 0 | 35269 | 391474 | 426743 |

| | |
|---|---|
| Correctly Classified Instances | 99.9902% |
| Incorrectly Classified Instances | 0.0098% |
| Dos Detection rate | 99.996% |
| False Alarm Ratio | 0.0074% |

Table 6: Training result of level 2 classifier

| Actual Class | Classified as | | | Total |
|---|---|---|---|---|
| | Normal | Other Attacks | Dos Attack | |
| Normal | 0 | 257 | 93 | 340 |
| Other Attacks | 0 | 3998 | 435 | 4433 |
| Dos Attack | **0** | **795** | **222524** | **223319** |
| Total | 0 | 5050 | 223042 | 228092 |

| | |
|---|---|
| Correctly Classified Instances | 99.3117% |
| Incorrectly Classified Instances | 0.6883% |
| Dos Detection rate | 99.644% |
| False Alarm Ratio | 0.2322% |

The third classifier is trained on only Probe, U2R and R2L attack data (refer to Table 7). The instances classified as Other Attacks by Level 2 classifier makes the testing data for third classifier. Table 8 shows the results of Level 3 classifier. The percentage detection rate of Probe attack is 100%.

Table 7: Training results of level 3 classifier

| Actual Class | Classified as | | | | Total |
|---|---|---|---|---|---|
| | Normal | Dos | Others | Probe | |
| Normal | 0 | 0 | 0 | 0 | 0 |
| Dos Attack | 0 | 0 | 0 | 0 | 0 |
| Other Attacks | 0 | 0 | 1172 | 6 | 1178 |
| Probe Attack | 0 | 0 | 3 | 4104 | 4107 |
| Total | 0 | 0 | 1175 | 4110 | 5285 |

| | |
|---|---|
| Correctly Classified Instances | 99.8297% |
| Incorrectly Classified Instances | 0.1703% |
| Probe Detection rate | 99.927% |
| False Alarm Ratio | 0.1459% |

The instances classified as Other Attacks by Level 3 classifier makes the testing data for this final level classifier. Level 4 classifier is trained on only U2R and R2L attack data (refer to Table 9). Table 10 shows the test results of level 4 classifier.

## 6.3 Anomaly Detection Results

The training data for CBA consist of whole of the 10% KDD training data available with normal and attack labels. It is the same training data as given to Level 1, C4.5 based decision tree classifier for training. The normal labeled output of Level 1 classifier is given as input to anomaly detector as test data.

Table 8: Confusion matrix of testing dataset for level 3 classifier

| Actual Class | Classified as | | | | Total |
|---|---|---|---|---|---|
| | Normal | Dos | Others | Probe | |
| Normal | 0 | 0 | 7 | 253 | 260 |
| Dos Attack | 0 | 0 | 471 | 358 | 829 |
| Other Attacks | 0 | 0 | 527 | 347 | 874 |
| **Probe Attack** | **0** | **0** | **0** | **3086** | **3086** |
| Total | 0 | 0 | 1005 | 4044 | 5049 |

| | |
|---|---|
| Correctly Classified Instances | 71.5587 % |
| Incorrectly Classified Instances | 28.4413 % |
| Probe Detection rate | 100% |
| False Alarm Ratio | 23.6894% |

Table 9: Confusion matrix of training dataset for level 4 classifier

| Actual Class | Classified as | | | | | Total |
|---|---|---|---|---|---|---|
| | Nor. | Dos | Probe | U2R | R2L | |
| Nor. | 0 | 0 | 0 | 0 | 0 | 0 |
| Dos | 0 | 0 | 0 | 0 | 0 | 0 |
| Probe | 0 | 0 | 0 | 0 | 0 | 0 |
| U2R | 0 | 0 | 0 | 45 | 7 | 52 |
| R2L | 0 | 0 | 0 | 5 | 1121 | 1126 |
| Total | 0 | 0 | 0 | 50 | 1128 | 1178 |

Nor.:Normal

| | |
|---|---|
| Correctly Classified Instances | 98.9813 % |
| Incorrectly Classified Instances | 1.0187 % |
| U2R Detection rate | 86.538% |
| R2L Detection rate | 99.555% |
| False U2R Alarm Ratio | 10.000% |
| False R2L Alarm Ratio | 0.6205% |

The advantage of taking both classes in training data is that only rules that differentiate normal profile from attack traffic will be generated based upon higher support and confidence value.

The values for minimum support (minsup) and minimum confidence (minconf) are chosen after few trials with different support and confidence values. It was found that using minsup as 3.00% and minconf as 100% profiling gives more accurate results.

Table 11 shows the confusion matrix of training data as given by CBA after training.

The rules generated (With MinSup: 3.000%, MinConf: 100.000%, RuleLimit: 80000) are like:

*Antecedents → Consequents (one of the target class)*

Table 10: Confusion matrix of training dataset for level 4 classifier

| Actual Class | Classified as | | | | | Total |
|---|---|---|---|---|---|---|
| | Nor. | Dos | Probe | U2R | R2L | |
| Nor. | 0 | 0 | 0 | 1 | 6 | 7 |
| Dos | 0 | 0 | 0 | 0 | 471 | 471 |
| Probe | 0 | 0 | 0 | 0 | 0 | 0 |
| **U2R** | **0** | **0** | **0** | **9** | **8** | **17** |
| **R2L** | **0** | **0** | **0** | **2** | **508** | **510** |
| *Total* | 0 | 0 | 0 | 12 | 993 | 1005 |

Nor.:Normal

| | |
|---|---|
| Correctly Classified Instances | 51.4428 % |
| Incorrectly Classified Instances | 48.5572 % |
| U2R Detection rate | 52.941% |
| R2L Detection rate | 99.607% |
| False U2R Alarm Ratio | 25.000% |
| False R2L Alarm Ratio | 48.8418% |

Table 11: Training results of cba classifier

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| *Normal* | 88419 | 8859 | 97278 |
| *Attack* | 0 | 396743 | 396743 |
| *Total* | 88419 | 405602 | 494021 |

| | |
|---|---|
| Correctly Classified Instances | 98.207% |
| Incorrectly Classified Instances | 1.793% |
| Anomaly Detection rate | 100.00% |
| False Alarm Ratio | 2.1841% |

(Cover% Conf% CoverCount SupCount Sup%).

The rule set generated from training data contains both types of rules; rules having consequents as normal class and rules having consequents as attack class.

These two types of rules are separated to make two different profiles: Normal Profile: Rules with consequent as normal class, default class as attack, minimum support as 3.000%, minimum confidence as 100.000% and RuleLimit as 80000.

**Normal Profile.** Rules with consequent as normal class, default class as attack, minimum support as 3.000%, minimum confidence as 100.000% and RuleLimit as 80000.

**Attack Profile.** Rules with consequent as Attack class, default class as normal, minimum support as 3.000%, minimum confidence as 100.000% and RuleLimit as 80000.

Some of the Attack profile rules are:

Table 12: Test result of classification using attack class profile (default class = normal)

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| Normal | 60252 | 1 | 60253 |
| Attack | 1867 | 11817 | 13684 |
| Total | 62119 | 11818 | 73937 |

| | |
|---|---|
| Correctly Classified Instances | 97.44 % |
| Incorrectly Classified Instances | 2.526 % |
| Anomaly Detection rate | 86.356% |
| False Alarm Ratio | 0.0084% |

- Rule1: dst_host_srv_diff_host_rate $< 0.005$, count $\geq 501.5 \rightarrow$ class = attack (53.145%, 100.000%, 262549, 262549, 53.145%).
- Rule 2: src_bytes = [1031.5, 1032.5), protocol_type = icmp $\rightarrow$ class = attack (46.155%, 100.000%, 228017, 228017, 46.155%).
- Rule 3: diff_srv_rate = [0.045, 0.085), src_bytes $< 2.5 \rightarrow$ class = attack (21.045%, 100.000%, 103967, 103967, 21.045%).
- Rule 4: dst_host_count $\geq 254.5$, src_bytes ¡ 2.5, service = 1 $\rightarrow$ class = attack (20.692%, 100.000%, 102224, 102224, 20.692%).

DefaultClass: class = normal.

Table 12 shows the test results when attack profile is used for classification. The false alarm ratio is as low as 0.0084% and Anomaly detection rate is 86.356%.
Some of the Attack profile rules are:

- Rule1: dst_host_rerror_rate ¡ 0.005, logged_in = 1, hot $< 0.5$, service = 3 $\rightarrow$ class = normal (10.873%, 100.000%, 53713, 53713, 10.873%)
- Rule 2: dst_host_srv_diff_host_rate = [0.005, 0.065), service = 3 $\rightarrow$ class = normal (6.545%, 100.000%, 32336, 32336, 6.545%)
- Rule 3: logged_in = 1, dst_bytes = [233.5, 1183.5), duration $< 0.5 \rightarrow$ class = normal (5.184%, 100.000%, 25609, 25609, 5.184%)
- Rule 4: src_bytes = [8.5, 17.5) $\rightarrow$ class = normal (0.095%, 100.000%, 469469, 0.095%)

DefaultClass: class = attack.

Table 13 shows the test result when normal profile is used for classification. The Anomaly Detection rate in this case is 99.021%.

Table 13: Test result of classification using normal class profile (default class = attack)

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| Normal | 52244 | 8049 | 60593 |
| Attack | 222 | 25024 | 250436 |
| Total | 52466 | 258563 | 311029 |

| | |
|---|---|
| Correctly Classified Instances | 90.08 % |
| Incorrectly Classified Instances | 9.92 % |
| Anomaly Detection rate | 99.021% |
| False Alarm Ratio | 26.2841% |

Table 14: Test result of overall integrated model

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| Normal | 52244 | 8009 | 60253 |
| Attack | 222 | 22462 | 22684 |
| Total | 52466 | 30471 | 82937 |

| | |
|---|---|
| Correctly Classified Instances | 90.08 % |
| Incorrectly Classified Instances | 9.92 % |
| Anomaly Detection rate | 99.021% |
| False Alarm Ratio | 26.2841% |

### 6.4 Results of Integrated Model

Our integrated model uses normal profile at anomaly detector to find deviations from normal behavior.

Table 14 shows the overall confusion Matrix for the integrated Model using Table 12 and Table 13. The false alarm ratio of the model is 3.229%.

Table 15 shows distribution of known and new attack patterns as detected by our integrated Model. Misuse detection rate is 99.995%. The anomaly detection rate is also as high as 99.298%.

Table 15: To-be tested audio files

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| Known attack | 12 | **220513** | 220525 |
| Anomaly | **210** | 29701 | 29911 |
| Total | 222 | 250214s | 250436 |

| | |
|---|---|
| Correctly Classified Instances | 90.08 % |
| Incorrectly Classified Instances | 9.92 % |
| Anomaly Detection rate | 99.021% |
| False Alarm Ratio | 26.2841% |

Table 16: KDD 99 10% training dataset and testing dataset distribution

| | Training Set | Testing Set |
|---|---|---|
| Normal | 19.69% | 19.48% |
| Probe | 0.83% | 1.34% |
| Dos | 79.24% | 73.90% |
| U2R | 0.01% | 0.07% |
| R2L | 0.23% | 5.20% |

## 7 New Improved Version of 10% Training Dataset and Experimental Results

KDD 99 Dataset has been reported with many deficiencies [16]. The KDD training dataset has huge number of similar records for Dos attack and normal traffic as compared to Probe, U2R and R2L attacks. However, the U2R and R2L attacks constitute 5.27% of the test dataset, which is a substantial increase compared to the training dataset. This causes the Level 1 classifier to be biased towards normal class and leads to high false negative count.

Table 16 compares the percentage distribution of instances in 10% training dataset with that in testing dataset. Distribution shows that Normal data and Dos instances make most of the training data. The other classes are just 1.07% in training data whereas they constitute 6.61% of total test data.

To deal with this problem, some authors [17] recommend removing the duplicated data from training dataset. Their technique can be good for comparing performance of different algorithms but it is hard to measure its effectiveness on real life scenario. In real life scenarios, traffic actually consists of such repeated patterns and it is important to make a classifier strongly learn such patterns. In this paper we suggest repetition of less frequent records. This duplication of data helps the classifier to learn less frequent patterns also with more efficiency.

For our experiments, we duplicated the U2R and R2L attack instances 100 times and Probe attacks 5 times. Only one copy of Dos and normal records was kept in the training set. We trained our first level classifier i.e. normal-attack classifier using this new dataset and tested this new classifier on earlier test data.

**Misuse and Anomaly Detection Results (On Improved KDD Dataset)**:

Table 17 shows the test results of Level 1 decision tree trained using new improved dataset. The results show that attack detection rate has increased from 90.942% to 92.251%. There is also an increase in overall accuracy from 92.5975% to 93.602%.

Table 18 shows the actual count of known and new attacks, and their percentage detection rate using classifier trained on original and new dataset.

Table 17: Test results of level 1 classifier after data duplication

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| Normal | 60099 | 494 | 60593 |
| Attack | 19405 | 231031 | 250436 |
| Total | 79504 | 231525 | 311029 |

| | |
|---|---|
| Correctly Classified Instances | 93.6022% |
| Incorrectly Classified Instances | 6.3978% |
| Attack Detection rate | 92.251% |
| False Alarm Ratio | 0.2133% |

The experimental results show that data duplication gives a high improvement in Level 1 classifier's misuse and anomaly detection rate from 99.6835% and 26.4618% to 99.9832% and 35.2479% respectively.

Table 19 shows the testing result of Anomaly Detector on the output (Normal traffic) of Level 1 decision tree trained on new improved Dataset. 52889 instances of total 52992 satisfy the normal user profile. Anomaly detector has a false alarm rate of 27.17% on this test dataset.

Table 19: Test result of anomaly detector trained on new improved dataset

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| Normal | 52889 | 7210 | 50099 |
| Attack | 88 | 19317 | 19405 |
| Total | 52992 | 26527 | 69519 |

| | |
|---|---|
| Correctly Classified Instances | 89.51 % |
| Incorrectly Classified Instances | 10.49 % |
| Attack Detection rate | 99.546% |
| False Alarm Ratio | 27.179% |

Table 20 gives the overall confusion matrix of Integrated Model trained on new improved Dataset. The result shows an improved overall accuracy of 97.495% compared to old dataset results of 97.24%.

# 8 Conclusions

In this paper we have developed new model for attack detection using a decision trees based sequential model for Misuse and CBA rules based classification model for Anomaly Detection. Model's performance is evaluated on DARPA KDD CUP99 benchmark. The overall accuracy of the proposed model is 97.24%, which is a drastic increase compared to the single C4.5 based classifier's accuracy of 92.59%. Misuse and anomaly detection of the integrated model is 99.995% and 99.298%, respectively.

Table 20: To-be tested audio files

| Actual Class | Classified as | | Total |
|---|---|---|---|
| | Normal | Attack | |
| Normal | 52889 | 7704 | 60593 |
| Attack | 88 | 250348 | 250436 |
| Total | 52977 | 258052 | 311029 |

| | |
|---|---|
| Correctly Classified Instances | 97.495 % |
| Incorrectly Classified Instances | 2.505 % |
| Attack Detection rate | 99.964% |
| False Alarm Ratio | 2.9854% |

The overall attack detection rate is 99.911%, and false alarm ratio of the integrated model is 3.229%. Also, individual attack detection rate of 99.644% for Dos and 100% for Probe, 52.941% for U2R and 99.607% for R2L attacks is achievable.

To overcome the deficiencies in KDD 99 dataset, a new improved dataset is also proposed. The new dataset has an improved overall accuracy of 97.495% compared to 97.24% of old dataset. The overall attack detection rate is 99.964% and the false alarm ratio is 2.985% on new dataset.

In the future, we are planning to create a new dataset of our own from live traffic. Then, the proposed model will be tested using C5.0 as the decision tree algorithm on the new dataset.

# References

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Proceedings of Intlernational Conference On Very Large Databases (VLDB)*, Sep. 1994.

[2] B. Asfa, D. Bekel, B. Eshete, A. Villaorita, and K. Weldemaria, "Host-based anomaly detection for pervasive systems," *Proceedings of 5th International Conference on Risks and Security of Internet and Systems*, 2010.

[3] S. Axellson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Transaction on Information and System Security*, vol. 3, no. 3, pp. 186-205, 2000.

[4] E. Bloedorn, L. Talbot, C. Skorupka, A. Christiansen, W. Hill, and J. Tivel, "Data mining applied to intrusion detection: MITRE experiences," *Proceedings of IEEE International Conference on Data Mining*, 2001.

[5] C5.0/See5. (http://rulequest.com/see5-comparison.html)

[6] V. Corey *et al.*, "Network forensics analysis," *IEEE Internet Computing*, vol. 6, no. 6, pp. 60V66, 2002.

[7] O. Depren, M. Topallar, E. Anarim, and M. K. Cili, "An intelligent intrusion detection system (IDS)

Table 18: Misuse and anomaly detection rate of level 1 classifier before and after data duplication

| **True Positives** | **Known Attacks** | **New Attacks** |
|---|---|---|
| In Test dataset | 220,525 | 29,911 |
| Detected by level 1 classifier (trained on original dataset) | 219,827 (99.6835 %) | 7,905 (26.4618%) |
| Detected by level 1 classifier (trained on new dataset) | 220,525 (99.9832%) | 10,543 (35.2479%) |

for anomaly and misuse detection in computer networks," *Expert Systems with Application*, vol. 29, no. 4, pp. 713-722, 2005.

[8] D. Hon, and L. Haibo, "A lightweight network intrusion detection model based on feature selection," *Proceedings of 15th IEEE Pacific Rim International Symposium on Dependable Computing*, 2009.

[9] S. S. Kandeeban and R. S. Rajesh, "Integrated Intrusion Detection System Using Soft Computing," *International Journal of Network Security*, vol. 10, no. 2, pp. 87-92, Mar. 2010.

[10] KDD Cup 1999 Data. (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)

[11] K. Kendall, *A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*, M. Eng. Thesis, Massachusetts Institute of Technology, Massachusetts, United States, June 1999.

[12] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," *Proceedings of Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, New York, USA, 1998.

[13] H. A. Nguyen and D. Choi, "Application of data mining to network intrusion detection: classifier selection model," *APNOMS 2008*, LNCS 5297, pp. 399-408, Springer-Verlag, 2008.

[14] Z. S. Pan, S. C. Chen, G. B. Hu, and D. Q. Zhang, "Hybrid neural network and C4.5 for misuse detection," *Proceedings of Second International Conference on Machine Learning and Cybernetics*, 2003.

[15] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114-132, 2007.

[16] M. Sabhnani and G. Serpen, "Why machine learning algorithms fail in misuse detection on KDD intrusion detection dataset," *Intelligent Data Analysis*, vol. 8, no. 4, pp. 403-415, 2004.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *Proceedings of IEEE Symposium*, 2009.

[18] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann, San Mateo, California, 1993.

[19] Weka V Data Mining Machine Learning Software. (http://www.cs.waikato.ac.nz/ml/weka/)

[20] C. Xiang, M. Z. Fang, and R. Chen, "An intrusion detection method for Ad hoc network based on class association rules mining," *Proceedings of International Conference on Frontier of Computer Science and Technology*, 2009.

**Radhika Goel** Integrated Dual Degree (B.Tech in Computer Science with M.Tech in Information Technology) final year student at the Department of Electronics and Computer Engineering IIT Roorkee, India. She is currently working on her Master's dissertation. Her areas of interest include Network Security, and Data Mining.

**Anjali Saradana** Phd in Electronics and Computer Engineering from IIT, Roorkee, India (2009). She is presently an Assistant Professor in the Department of Electronics and Computer Engineering IIT Roorkee, India. Areas of Interest include Information and Network Security, Wireless Security and Intrusion Detection. Dr. Sardana has been working in the area of security and honeypots for the past 7 years. She has delivered several expert lectures at faculty training programs, seminar and workshops, and has published articles in international journals. She has received several awards incuding Google Women in Engineering Award and Government of India R&D award.

**Ramesh C. Joshi** Phd in Electronics and Computer Engineering from University of Roorkee, India(1980). Teaching for the past 42 years and is presently Professor in the Department of Electronics and Computer Engineering, IIT Roorkee, India. Dr. Joshi has been involved in wide spectrum of fields like Parallel and Distributed Processing, Data mining, Information Systems, Bioinformatics, Information Security and Digital Forensics.

He has chaired several conferences and has delivered various special lectures. He was awarded the prestigious Gold Medal by Institute of Engineers (India) in 1978. He was also a member of National Industrial Research and Development Award Committee. He is also Chairman of Planning and Curriculum Development Ambedkar University, Lucknow and various MIT and AICTE committes.