Classifying the Network Intrusion Attacks using Data Mining Classification Methods and their Performance Comparison

P Srinivasulu[1], D Nagaraju[2], P Ramesh Kumar[3], and K Nageswara Rao[4]

Abstract
Security is becoming a critical part of organizational information systems. Intrusion Detection System (IDS) is an important detection that is used as a countermeasure to preserve data integrity and system availability from attacks. The main reason for using Data Mining Classification Methods for Intrusion Detection Systems is due to the enormous volume of existing and newly appearing network data that require processing. In this paper we are using CART [1] [4], Naïve Bayesian [2] [10], and Artificial Neural Network Model [3] [10], data mining classification methods. These are proving to be useful for gathering different knowledge for Intrusion Detection. This paper presents the idea of applying data mining classification techniques to intrusion detection systems to maximize the effectiveness in identifying attacks, thereby helping the users to construct more secure information systems.

Keywords:
Data mining, Information Security, Intrusion Detection, Classification, Confusion Matrix

1. Introduction

Computer based Information Systems are becoming an integral part of our organizations. An Information System is a computerized system which contains organization information which serves the organization in its various activities and functions. Computer Security is the ability to protect a computer system and its resources with respect to confidentiality, integrity, and availability. Various protocols, firewalls are in existence to protect these systems from computer threats. Intrusion is a type of cyber attack that attempts to bypass the security mechanism of a computer system. Such an attacker can be an outsider who attempts to access the system, or an insider who attempts to gain and misuse non-authorized privileges.
Data Mining is assisting various applications for required data analysis. Recently, data mining is becoming an important component in intrusion detection system. Different data mining approaches like classification, clustering, association rule, and outlier detection are frequently used to analyze network data to gain intrusion related knowledge. In this paper we elaborate on several of these data mining classification techniques and will describe how they are used in the classification of intrusion detection attacks.
Data Mining is an analytic process designed to explore data (usually large amounts of data - typically business or market related) in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. The ultimate goal of data mining is prediction. Predictive data mining is the most common type of data mining and one that has the most direct business applications. The process of data mining consists of three stages: (1) the initial exploration, (2) model building or pattern identification with validation/verification, and (3) deployment (i.e., the application of the model to new data in order to generate predictions). These steps are explained below

Stage 1: Exploration. This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records and - in case of data sets with large numbers of variables ("fields") - performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods which are being considered). Then, depending on the nature of the analytic problem, this first stage of the process of data mining may involve anywhere between a simple choice of straightforward predictors for a regression model, to elaborate exploratory analyses using a wide variety of graphical and statistical methods (see *Exploratory Data Analysis (EDA)*) in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

Stage 2: Model building and validation. This stage involves considering various models and choosing the best one based on their predictive performance (i.e., explaining the variability in question and producing stable results across samples). This may sound like a simple operation, but in fact, it sometimes involves a very elaborate process. There are a variety of techniques developed to achieve that goal - many of which are based on so-called "competitive evaluation of models," that is, applying different models to the same data set and then comparing their performance to choose the best. These techniques - which are often considered the core of predictive data mining - include: Bagging (Voting, Averaging), Boosting, Stackin (Stacked Generalizations), and Meta-Learning.

Manuscript received June 5, 2009
Manuscript revised June 20, 2009

Stage 3: Deployment. That final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

1.1 Data Mining Models

In the business environment, complex data mining projects may require the coordinate efforts of various experts, stakeholders, or departments throughout an entire organization. In the data mining literature, various "general frameworks" have been proposed to serve as blueprints for how to organize the process of gathering data, analyzing data, disseminating results, implementing results, and monitoring improvements.
One such model, **CRISP** (Cross-Industry Standard Process for data mining) [5] was proposed in the mid-1990s by a European consortium of companies to serve as a non-proprietary standard process model for data mining. This general approach postulates the following (perhaps not particularly controversial) general sequence of steps for data mining projects shown below in Figure 1.

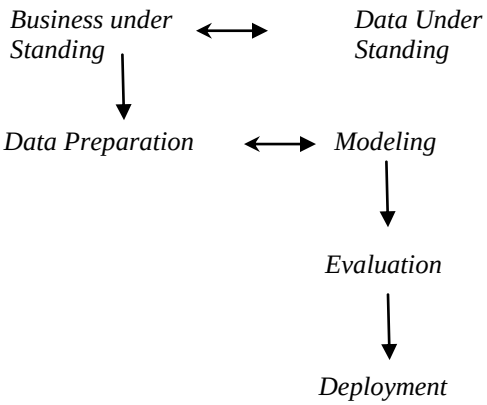

Figure 1. CRISP Methodology

Another approach - the **Six Sigma** [6] methodology is a well-structured, data-driven methodology for eliminating defects, waste, or quality control problems of all kinds in manufacturing, service delivery, management, and other business activities. This model has recently become very popular (due to its successful implementations) in various American industries, and it appears to gain favor worldwide. It postulated a sequence of, so-called, **DMAIC** steps shown below:

Define → Measure → Analyze → Improve → control

2. Problem Statement

In this section, an overview of the proposed framework is given below. Then we will illustrate how to apply the classification mining techniques for detecting the network intrusions.

2.1 Overview of the Framework

Our current architecture for intrusion detection is shown in Figure 2. Network traffic is analyzed by a variety of available sensors. This sensor data is pulled periodically to a central server for conditioning and input to a relational database. HOMER filters events from the sensor data before they are passed on to the classifier analyses. Data mining tools filter false alarms and identify anomalous behavior in the large amounts of remaining data. A web server is available as a front end to the database if needed, and analysts can launch a number of predefined queries as well as free form SQL queries from this interface. The goal of this operational model is to have all alarms reviewed by human analysts.

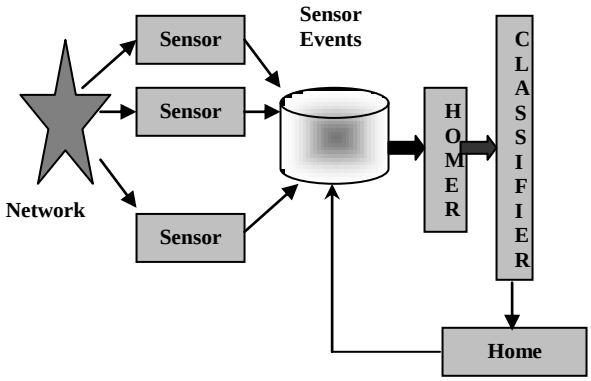

Figure 2. How Sensors feed into Intrusion Detection System

3. Building Classification Models

In this paper we are given below the details of Tree based model, Bayesian model, and Neural Network based model.

3.1 Induction Decision Tree based Model

Induction Decision Tree or simply *the Classification trees* are used to predict membership of cases or objects in the classes of a categorical dependent variable from their measurements on one or more predictor variables. *Classification tree* analysis is one of the main techniques used in so-called *Data Mining*. Classification and regression trees (CART) [10] is a non-parametric technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively.
Trees are formed by a collection of rules based on values of certain variables in the modeling data set. Rules are selected based on how well splits based on variables' values can differentiate observations based on the dependent variable. Once a rule is selected and splits a

node into two, the same logic is applied to each "child" node (i.e. it is a recursive procedure)
Splitting stops when CART detects no further gain can be made, or some pre-set stopping rules are met. Each branch of the tree ends in a terminal node. Each observation falls into one and exactly one terminal node. Each terminal node is uniquely defined by a set of rules For example, we might have a decision tree to help a financial institution decide whether a person should be offered a loan, which is shown in figure3.

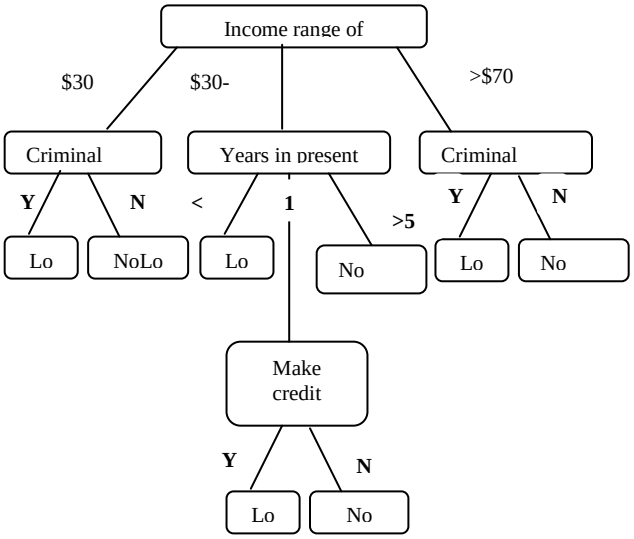

Figure 3. Inductin tree

3.2 Bayesian Model

There are many variations of the Bayesian models are available. These are based on the Bayes theorem. One is the Na ïve Bayesian model which is described below.

i) Let D be a training set of tuples and their associated class labels. Each tuple is represented by an n-dimensional attribute vector, X = (x_1, x_2, ……, x_n), depicting n

measurements made on the tuple from n attributes, respectively, A_1, A_2, …., A_n.

ii) Suppose that there are m classes, C_1,C_2,…….C_m. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the na ïve Bayesian classifier predicts that the tuple X belongs to the class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X)$$
$$\text{for} \quad 1 \le j \le m, j \ne i.$$

Thus we maximize P(Ci/X). The class Ci for which P(Ci/X) is maximized is called the maximum posterior hypothesis. By Bayes's theorem Equation (1)

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \qquad (1)$$

iii) As P(X) is constant for all cases, only $P(X/C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) =\ldots\ldots P(C_m)$, and we therefore maximize P(X/Ci). Otherwise, we maximize $P(X/C_i)P(C_i)$.

iv) Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X/C_i)$. In order to reduce computation in evaluating $P(X/C_i)$, the na ïve assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given that class label of the tuple. Thus

$$P(X / C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) \qquad (2)$$
$$= P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots\ldots\ldots\ldots \times P(x_n \mid C_i).$$

We can easily estimate the probabilities $P(x_1/C_i)$, $P(x_2/C_i)$,….. $P(x_n/C_i)$ from the training tuples.
a). If A_k is categorical, then $P(x_k/C_i)$ is the number of tuples of class Ci in D having the values x_k for A_k, divided by $|C_i, D|$, the number of tuples of class C_i in D.
b). If A_k is continuous-valued, then we need to do a bit more work, but the calculation is pretty straightforward. A continuous-valued attribute [11] is typically assumed to have a Guassian distribution with mean and standard deviation

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (3)$$

3.3 Artificial Neural Network Model

An artificial neural network is a system based on the operation of biological neural networks, in other words, is an emulation of biological neural system. Why would be necessary the implementation of artificial neural networks? Although computing these days is truly

advanced, there are certain tasks that a program made for a common microprocessor is unable to perform
To capture the essence of biological neural systems, an artificial *neuron* is defined as follows:

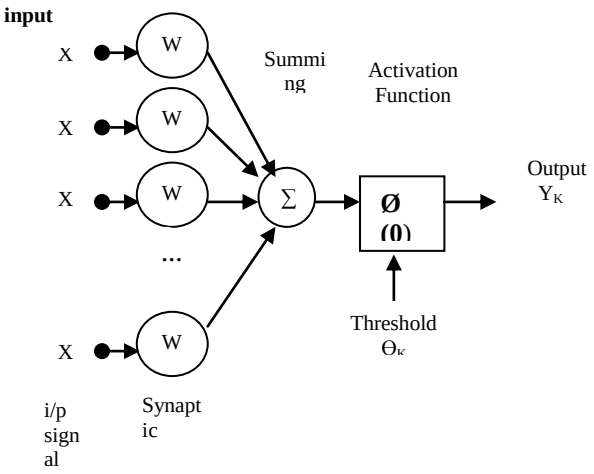

Figure 4. Basic ANN Model

It receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection that has a strength (or *weight*); these weights correspond to synaptic efficacy in a biological neuron. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold subtracted, to compose the *activation* of the neuron (also known as the post-synaptic potential, or PSP, of the neuron). From this model the interval activity of the neuron can be shown to be:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j \qquad (3)$$

The activation signal is passed through an activation function (also known as a transfer function) to produce the output of the neuron.
If the step activation function is used (i.e., the neuron's output is 0 if the input is less than zero, and 1 if the input is greater than or equal to 0) then the neuron acts just like the biological neuron described earlier (subtracting the threshold from the weighted sum and comparing with zero is equivalent to comparing the weighted sum to the threshold). Actually, the step function is rarely used in artificial neural networks, as will be discussed. Note also that weights can be negative, which implies that the synapse has an inhibitory rather than excitatory effect on the neuron: inhibitory neurons are found in the brain.
The activation function as mentioned previously, the activation function acts as a squashing function, such that the output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). In general, there are three types of activation functions, denoted by Φ(.) . First, there is the Threshold Function which takes on a value of 0 if the summed input is less than a certain threshold value (v), and the value 1 if the summed input is greater than or equal to the threshold value.

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

4. Dataset Description

We conducted experiments on KDDCup99 dataset [4]. The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes.
Attacks fall into four main categories:

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow'' attacks;
- Probing: surveillance and other probing, e.g., port scanning.

The dataset contain a total of 65 attributes in which there are 24 training attack types, with an additional 14 types in the test data only. Some of the attribute names with their type were listed below:

Duration	: continuous.
protocol_type	: symbolic.
Service	: symbolic.
Flag	: symbolic.
src_bytes	: continuous.
dst_bytes	: continuous.
Land	: symbolic.
wrong_fragment	: continuous.

Urgent : continuous.
Hot : continuous.
num_failed_logins: continuous.
logged_in : symbolic.
num_compromised: continuous.
root_shell : continuous.

5. Experiments and Performance Evaluation

For experiments we used Weka3.6 data mining tool [4] for analyzing the results. The classification models can be evaluated using misclassification error rate and the area under ROC curve. These explained below

(a) Confusion Matrix

One of the methods to evaluate the performance of a classifier is using confusion matrix. A Confusion matrix that summarizes the number of instances predicted correctly or incorrectly by a classification model.

The *confusion matrix* [14] is more commonly named *contingency table* which is shown in Table1. For example we have two classes + and -, and therefore a 2x2 confusion matrix, the matrix could be arbitrarily large. The number of correctly classified instances is the sum of diagonals in the matrix; all others are incorrectly classified (class ”a” gets misclassified as ”b” exactly twice, and class ”b” gets misclassified as ”a” three times). The following terminology is often used when referring to the counts tabulated in a confusion matrix.

		Predicted Class	
		+	-
Actual Class	+	TP	FN
	-	FP	TN

Table 1: A confusion matrix for a binary classification problem in which the classes are not equally important

(i) The **True Positive (TP)** [12]**:** corresponds to the number of positive examples correctly predicted by the classification model.

(ii) The **False Negative (FN)** [12]**:** corresponds to the number of positive examples wrongly predicted as negative by the classification model.

(iii) The **False Positive (FP)** [12]**:** corresponds to the number of negative examples wrongly predicted as positive by the classification model.

(iv) The **True Negative (TN)** [12]**:** corresponds to the number of negative examples correctly predicted by the classification model.

The counts in a confusion matrix can also be expressed in terms of percentages. The true positive rate (TPR) [12] [9] or sensitivity is defined as the fraction of positive examples predicted correctly by the model, i.e.,

$$TPR = TP / (TP + FN) \quad (3)$$

Similarly, the true negative rate (TNR) [12] [9] is defined as the fraction of negative examples predicted correctly by the model, i.e.,

$$TNR = TN / (TN + FP) \quad (4)$$

False positive rate(FPR) is the fraction of negative examples predicted as a positive class, ie.,

$$FPR = FP / (TN + FP) \quad (5)$$

Finally the false negative rate (FNR) [12] [9] is the fraction of positive examples predicted as a negative class. ie.,

$$FNR = FN / (TP + FN) \quad (6)$$

(v) **Recall and Precision:** are two widely used metrics employed in applications where successful detection of one of the classes is considered more significant than detection of the other classes [14]. A formal definition of these metrics is given below. The Recall and Precision values are shown in the tables 4, 5, and 6 for each classification model.

$$\text{Precision, } p = \frac{TP}{TP+FP} \quad (7)$$

$$\text{Recall, } r = \frac{TP}{TP+FN} \quad (8)$$

(b) The Receiver Operating Characteristic Curve (ROC)

A Receiver Operating Characteristic curve(ROC) (Zweig, 1993) [11][13] summarizes the performance of a two-class classifier across the range of possible thresholds. It plots the sensitivity (class two true positives) versus one minus the specificity (class one false negatives). An ideal classifier hugs the left side and top side of the graph, and the area under the curve is 1.0. A random classifier should achieve approximately 0.5 (a classifier with an area less than 0.5 can be improved simply by flipping the class assignment). The ROC curve is recommended for comparing classifiers, as it does not merely summarize performance at a single arbitrarily selected decision threshold, but across all possible decision thresholds.

The ROC curve can be used to select an optimum decision threshold. This threshold (which equalizes the probability of misclassification of either class; i.e. the probability of false-positives and false-negatives) can be used to automatically set confidence thresholds in classification

networks with a nominal output variable with the Two-state conversion function.
ROC curve is a graphical approach for displaying the tradeoff between true positive rate and false positive rate of a classifier. In an ROC curve, the true positive rate (TPR) is plotted
along the y axis and the false positive rate (FPR) is on the x axis. Each point along the curve corresponds to one of the models induced by the classifier. The values of these are listed in the tables 4, 5, and 6 shown below for each classification model.

6. Conclusion

In this paper we applied the classification methods for classifying the attacks (intrusions) on DARPA dataset. Here we have not used some of the attributes, because the Weka3.6 does not support large databases. The results showing the performance of the Induction tree method and ANN methods are better than the NB classifier. But the time taken is more for ANN than other classifiers. We can extend this experiment for SVM and Multilayer ANN which are better classifier.

	Induction Tree	**Naïve Bayesian**	**ANN**
Total Number of Instances	19870	19870	16660
Correctly Classified Instances	99.839/%	95.8245 %	99.4118%
Incorrectly Classified Instances	0.1608%	4.1755 %	0.5882 %
Mean absolute error	0.0003	0.0049	0.0009
Root mean squared error	0.0137	0.0673	0.0237
Relative absolute error	0.3506 %	6.7775 %	1.3261 %
Root relative squared error	7.247 %	35.4803 %	12.9796 %

Table 2: Comparison of the three models

TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
.999	.002	.998	.999	.999	.999	Normal
1	0	.999	1	.999	1	Neptune
.999	0	1	.999	.999	1	Smurf
.952	0	.952	.952	.952	0.964	guess_passwd.
1	0	1	1	1	1	pod.
1	0	1	1	1	1	teardrop.
.733	0	.917	.733	.815	1	portsweep
.977	0	.986	.977	.981	.988	ipsweep
0	0	0	0	0	?	land
0	0	0	0	0	.744	ftp_write
1	0	1	1	1	1	back
0	0	0	0	0	?	imap
0	0	0	0	0	.5	buffer_overflow
0	0	0	0	0	?	satan
0	0	0	0	0	.5	phf
.911	0	.976	.911	.943	.996	nmap
0	0	0	0	0	.997	multihop

Table 4: Induction Tree Model summary report

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
.931	.003	.997	.931	.963	.999	Normal
.999	0	1	.999	.999	1	Neptune
.994	0	.998	.994	.996	1	Smurf
.952	.002	.4	.952	.563	0.964	guess_passwd.
1	0	.667	1	.8	1	pod.
1	.011	.13	1	.23	1	teardrop.
.933	0	.636	.933	.757	1	portsweep
.94	.019	.346	.94	.506	.988	ipsweep
0	0	0	0	0	?	land
0	.001	.05	.25	.083	.744	ftp_write
.25	0	1	.975	.987	1	back
.975	0	0	0	0	?	imap
0	0	0	0	0	.5	buffer_overflow
0	0	0	0	0	?	satan
0	0	0	0	0	.5	phf
.911	.004	.32	.911	.474	.997	nmap
0	.002	0	0	0	.997	multihop

Table 5: Naïve Bayesian Model summary report

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
.931	0	1	1	1	1	Smurf
.999	.001	1	.993	.996	1	Normal
.994	0	1	.941	.97	1	guess_passwd
.952	0	.5	1	.667	1	pod
1	0	.914	1	.955	1	teardrop..
1	0	.889	1	.941	1	portsweep
.933	.001	.923	.931	.927	1	ipsweep
.94	0	0	0	0	.987	land
0	.003	.067	1	.125	.999	ftp_write
0	0	1	1	1	1	back
.25	0	0	0	0	?	imap
.975	0	0	0	0	?	buffer_overflow
0	0	0	0	0	.99	satan
0	0	0	0	0	.861	phf
0	.001	.733	.917	.815	.999	nmap
.911	.001	.1	.5	.167	.994	multihop
0	0	1	1	1	1	neptune.

Table 6: Artificial Network Model summary report

References

[1] J. R Quilnlan C4.5: Programs for Machine Learning . Morgan-Kaufmann Publishers, San Mateo, CA 1993

[2] Rish, Irina. (2001). "An empirical study of the naive Bayes classifier". IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.

[3] M. Ramoni and P Sebastiani. Robust Bayes classifiers. Artificial Intelligence, 125: 209-226, 2001

[4] L. Breiman, J. H. Freidman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Belmont, CA: Wadsworth, 1984.

[5] U. Rajendra Acharya, P. Subbanna Bhat, S. S. Iyengar, Ashok Rao and Sumeet Dua *Pattern Recognition, Volume 36, Issue 1, January 2003, Pages 61-68*

[6] KDDCup99 datasets, The UCI KDD Archive: http:/kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[7] Henk de Koning, Jeroen de Mast, International Journal of Quality & Reliability Management, Vol. 23, Issue 7, page: 766-787

[8] www.crisp-dm.org/CRISPWP-0800.pdf

[9] Tan Michael Steinbach Vipin Kumar. Intruduction to Data Mining, Pang-Ning, Pearson Education, 2007.

[10] W. W Cohen. Fast Effective Rule Induction. In Proc. Of the 12th Intl. conf. on Machine Learining , pages 115-123, Tahoe city, CA, July 1195

[11] J. Han and M. Kamber. Data Mining : Concepts and Techiniques. Morgan Kaugmann Publishers, San Francisco, 2001.

[12] C. Ferri, P. Flach, and J. Hernandez-Orallo. Learning 'Decision Trees Using the Area Under the ROC Curve. In Proc. Of the 19th Intl. conf. on Machine Learning , pages 139-146, Sydney, Australia, July 2002

[13] A. P. Bradley. The use of the area under the ROC curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, 30(7):1145-1149, 1997.

[14] M. V. Joshi. On Evaluating Performance of Classifiers for Rare Classes. In Proc. Of the 2002 IEEE Intl. conf. on Data Mining, Maebashi City, Japan, December 2002.

P. Srinivasulu received his B. Tech from Acharya Nagarjuna university, Guntur, Andhra Pradesh in 1994 and completed post graduation from Jawaharlal Nehru Technological University, Hyderabad in 1998. He is currently pursuing Ph. D from Acharya Nagarjuna University, Guntur and working as Assistant Professor in V R Siddhartha Engineering College, in the Department of Computer Science and Engineering, Vijayawada, Andhra Pradesh. His research interests include Data Mining and Data Warehousing, Computer Networks, Network security and Parallel Computing. He has more than thirteen years of experience in teaching in many subjects, industry and in research. He is the member of Indian Society of Technical Education (ISTE) and also member of Computer Society of India (CSI). He has many publications in National and International conferences. He was selected for the Journal of Who is who.

D. Naga Raju received his B. Tech from Sri Venkateswara university, Tirupathi, Andhra Pradesh in 2002 **,** M.Tech from Jawaharlal Nehru University (JNU), NewDelhi in 2005. He is currently pursuing Ph.D from Jawaharlal Nehru Technological University, Hyderabad.He is prsently Assistant Professor in Department of Computer Science and Engineering, Acharya Nagarjuna University, Nagarjuna Nagar, Guntur Andhra Pradesh. His research interests include Image Processing, Pattern Recognition and Data Mining. He has more than Five years of experience in teaching and in research. He has published many papers in National and International conferences.

P.Ramesh Kumar received B.Tech (Computer Science and Engineering) Degree from Pondicherry University, in 2004 and M.E. degree from Sathyabama University in 2007.He is currently serving as Sr.Lecturer in the Department of Computer Science and Engineering, V.R.Siddhartha Engineering College. His research interest lies in the area of Ear Biometrics and Cryptography, Parallel Computing and Key Management. He is the member of Indian Society of Technical Education (ISTE) and also member of Computer Society of India (CSI). He has many publications in National and International conferences.

Dr. K N Rao, received his B. E in Electronics and munication Communication Engineering from Karnataka University, Bangalore. He was completed post graduation in Computer Science and Engineering, and Doctorate degree in Computer Science from Andhra University, Visakhapatnam, after serving many years in Andhra University, took volunteer retirement and now serving as a Professor and Head of the Department of Computer Science and Engineering of P V P Siddhartha Institue of Technology, Vijayawada, AP. Research interests include cryptography, and robotics. He is a member of many professional societies like, IEEE, ACM, IE, IETE, CSI, and ISTE. He is an National Board of Accreditation(NBA) of All India Council for Technical Education (AICTE), New Delhi expert committee member, and member of Post Graduation Board of studies Andhra University for M. Sc(Tech), and Very Large Scale Integration(VLSI) and Digital Signal Processing.