Distributed Data Mining Approach to Credit Card Fraud Detection

Dipti Thakur and Shalini Bhatia

Abstract - **The detection of fraudulent transactions in credit card world is an important application of classification techniques. As human behavior is unpredictable classifying any transaction either[1] as fraud or non-fraud is not acceptable. The paper shows how credit card transactions can be classified in various fraud levels depending on different fraudulent situations mined from the historical behavior of the customers. The technique used to perform classification is decision tree methodology of data mining. Also it is shown that how agent based classification can be used to share the rules among different credit card companies without sharing the data. It also helps to design a classifier which scales well & handles data of large magnitude.**

Index Terms - **Agent Based Classification, Data Mining, Decision Tree**

I. INTRODUCTION

A secured and trusted inter-banking network [1], [2] for electronic commerce requires high speed verification and authentication mechanisms that allow legitimate users easy access to conduct their business, while thwarting fraudulent transaction attempts by others. Fraudulent electronic transactions are already a significant problem, one that will grow in importance as the number of access points in the nation's financial information system grows.

Financial institutions today typically develop custom fraud detection systems targeted to their own asset bases. Most of these systems employ some machine learning and statistical analysis algorithms to produce *pattern-directed inference systems* [1]. They use models of anomalous or errant transaction behaviors to forewarn of impending threats.

These algorithms require analysis of large and inherently distributed databases of information about transaction behaviors to produce models of "probably fraudulent" transactions. Recently banks have come to realize that a unified, global approach is required, involving the periodic sharing of information about attacks with each other. Such information sharing is the basis of building a global fraud detection infrastructure where local detection systems propagate attack information to each other, thus preventing intruders from disabling the global financial network.

As credit card transactions continue to grow in number, taking an ever-larger share of the country's banking system and leading to a higher rate of stolen account numbers and subsequent losses by banks [3], improved fraud detection thus has become essential to maintain the viability of the banking system. Large-scale data-mining techniques [4] can improve on the state of the art in commercial practice. Scalable techniques to analyze massive amounts of transaction data that efficiently compute fraud detectors in a timely manner is an important problem, especially for e-commerce. Besides scalability and efficiency, the fraud-detection task exhibits technical problems that include skewed distributions of training data [2], [5] and non-uniform cost per error, both of which have not been widely studied in the knowledge-discovery and data mining community.

The system approach addresses the efficiency and scalability issues in several ways. In this a large data set of labeled transactions (either fraudulent or legitimate) is divided into smaller subsets, & mining techniques have been applied to generate classifiers in parallel, and resultant base models have been combined by meta-learning [1], [2], [5] from the classifier's behavior to generate a meta-classifier[1], [2], [5].

II. DESCRIPTION

The Overall Architecture of Distributed Fraud Detection System is shown in Fig. 1. Data Sites are the local data stores where the local classifiers will be derived. Classification engine acts as a re-combination agent to build a unified global classifier. The detection engine enables one to identify the fraudulent behavior of every input. The detection engine can be based on an online model working in real time, or in an offline manner working on stored, human assisted manner.

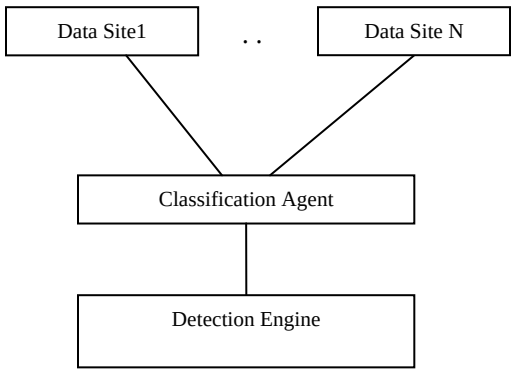

Fig.1 Overall Architecture of Distributed Fraud Detection System

Here, the approach is to use the decision tree classification mechanism to build local classifiers. The two main steps for this are:

Identify a meta-association between several local classifiers.

Dipti V. Thakur is an M.E. (Computer Engineering) student of Thadomal Shahani Engineering College, Mumbai, Maharashtra, India (e-mail: dipti_vt@yahoo.com).

Shalini Bhatia is the Head, Computer Engineering Department, Thadomal Shahani Engineering College, Mumbai, Maharashtra, India (e-mail: shalini.tsec@gmail.com).

Unify these to yield a global classifier.

The system has two key component technologies: local fraud detection agents that learn how to detect fraud and provide intrusion detection services within a single corporate information system, and a secure, integrated meta-learning system that combines the collective knowledge acquired by individual local agents. Once derived local classifier agents (models, or base classifiers) [1], [2] are produced at some site(s), two or more such agents may be composed into a new classifier agent (a *meta-classifier*) by a meta-learning agent [1], [2]. Meta-learning is a general strategy that provides the means of learning how to combine and integrate a number of separately learned classifiers or models; a meta-classifier is thus trained on the correlation of the predictions of the *base classifiers*. The meta-learning system proposed will allow financial institutions to share their models of fraudulent transactions by exchanging classifier agents in a secured agent infrastructure. But they will not need to disclose their proprietary data. In this way their competitive and legal restrictions can be met, but they can still share information. The detailed architecture of the system is explained below.

A. Data Cleansing module

The input data given to the classifier for learning is in the form of credit card transactions. The transactions used for training are comprised of fields having current transaction data combined with attributes representing some historical information about credit card customer behaviour. As due to privacy constraints bank has provided only summarized information of the credit card database, the information is rearranged and data is cleaned. Data cleaning means only the attributes giving information about fraud situations are picked up. Many of the attributes having continuous values are discretized for the implementation purpose. Also, ranges of the different attribute values are decided. The data received from the bank were included i.e. card holder's profile, showing its personal, educational & economical status & purchase profile giving review of its purchasing behavior within a year.

B. Data Site Architecture

As the system deals with distributed environment, there are different local sites having their on local credit card database of particular branch. The architecture of the data site is depicted in Fig 2. Each data site is deployed with the classification model having decision tree learning & classification algorithm. The classifier is trained with the local database to form the decision rules. These rules are then used for further classification.

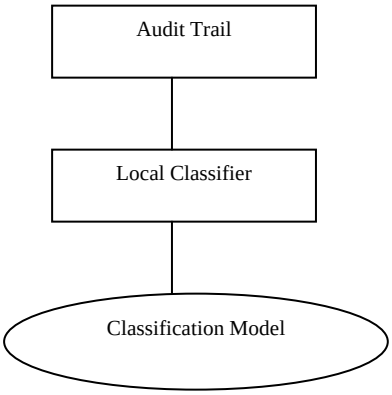

Fig. 2 Data Site architecture

C. Classification Agent Architecture

There can be various fraudulent situations and may be local to particular branch. So if any outlier behaviour takes place at a particular branch it will go undetected due to local classifier having rules related to local database. To overcome this situation agent architecture is developed and shown in Fig. 3.

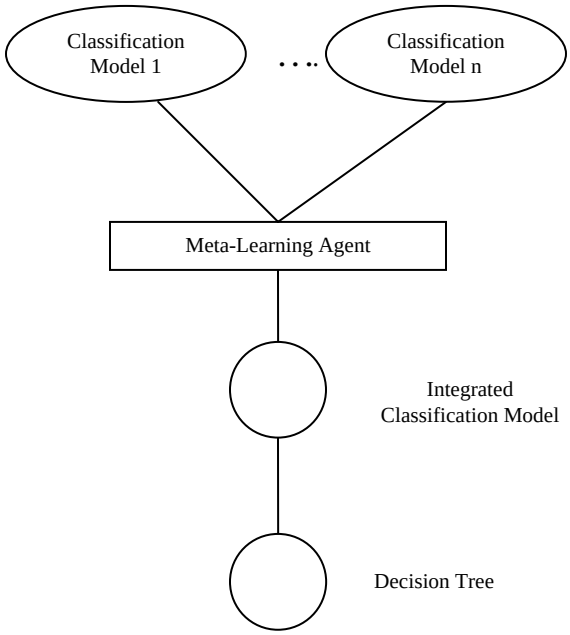

Fig. 3 Classification Agent architecture

Decision rules formed based on local database is shared among different data sites; the task of transferring these decision rules among different data sites is done by the agent architecture. Meta learning agent is used to combine the rules from all the data sites to form a single global decision tree which would be useful for taking global decisions.

D. Detection Engine Architecture

After classification model is developed detection engine shown in fig. 4 is used to classify current credit card transaction and returns the class level of the given transaction.

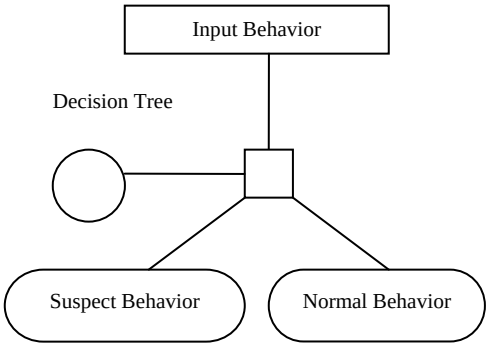

Fig4. Detection Engine Architecture

Here, detection engine works as front end where the transactions are given as input, the decision rules formed will be applied to classify the given input behavior in four different levels out of which level1 is normal & rest three are suspicious.

E. Tree Pruning

With decision tree classification algorithm, when decision tree is formed sometimes it happens that it generates some unwanted & meaningless rules as it grows deeper, it is called as overfitting [6]. This can be avoided by only considering those attributes which will have big contribution in forming the particular rule. This is done by stopping the growth of decision tree at particular level so that the rules formed give better classification.
Pre- pruning [6], [7] method is applied in the developed application. In this, the growth of decision tree is stopped at particular level & the remaining transactions at that particular split node are assigned with a most frequent class among these transactions.

III. IMPLEMENTATION

A. Credit Card Database

The credit card database used for training & classification is developed based on the snapshot of the credit card database given by the Bank. For security purpose the bank did not reveal the real data, due to this the database was designed manually from the given information and the overall survey of the credit card world. The credit card transaction table built for learning contains 101580 records.

The transaction table is built based on the current transaction information like amount, transaction time, transaction location, expiry date entered, card limit, in addition to that some historical information is also combined with these fields like average purchase of previous three months, average purchase of previous twelve months, customer's preferred transaction location and time, limit of number of transactions within a day to trace the customer's normal behavior. The transaction record does not contain customer account number because instead of learning, behavior models of individual customer accounts, overall models that try to differentiate legitimate transactions from fraudulent ones is built. So the model is customer-independent.

B. Types of Fraud

Instead of classifying the given transactions in only two types that is either fraud or non-fraud, in the system implemented transaction gets classified in four different types of class levels (L1, L2, L3, L4) which are decided based on different fraudulent situations traced out from given snapshot of database by bank & survey done on credit card world. The fraudulent situations based on which class levels have been assigned to the transactions.

C. Decision Tree Induction Algorithm

For the local classification ID3 [8] algorithm is implemented with added features of C4.5 [9], [10] algorithm. Also ID3 is combined with pre-pruning to increase the accuracy of classification.

ID3 Algorithm with pre-pruning

a) Construct the tree in a top-down recursive divide-and-conquer manner.
b) In the beginning, keep all the training examples at the root.
c) Attributes are considered to be categorical (if continuous-valued, they are discretized in advance).
d) Partition examples recursively based on selected attributes.
e) Select the splitting attribute on the basis of entropy measure.
f) Repeat all the steps until one of the three conditions get satisfied:
    i. All samples for a given node belong to the same class.
    ii. There are no remaining attributes for further partitioning.
    iii. There are no samples left.
    iv. Set prune level is reached.

Entropy Measure
Entropy measure [8] is given by following equation
For a set of record S

$$\text{Entropy } E(S) = -\Sigma\, p_j \log p_j \quad (j= 1\ldots m) \qquad (1)$$

Where p_j is the relative frequency of class j in S

Entropy divides S with n records in two sets, S1 with n1 records and S2 with n2 records.

$$E(S1,S2) = \frac{n_1}{n} E(S1) + \frac{n_2}{n} E(S2) \qquad (2)$$

In the context of decision trees, if the outcome of a node is to classify the records into two classes, C1 and C2, the outcome can be viewed as message that is being generated and the

entropy gives the measure of information for a message to be C1 or C2. If a set of records T is partitioned into a set of disjoint exhaustive classes C1,C2,…., Cn on the basis of a value of the class attribute, then the information needed to identify the class of an element of T is

Info (T) = Entropy (P), (3)

Where, P is probability distribution of the partition C1, C2, …,Cn. P is computed based on their relative frequencies, i.e.,

P = ((|C1|/|T|, |C2|/|T| , …|Cn|/|T|) (4)

The goal is to lower the Entropy.

D. Classification Algorithm

There are two phases in decision tree classification, first is to generate the decision tree from the given training data and second is actual classification where decision rules of formed decision tree is applied to the transaction having unknown class label to classify it in one of the classes. The algorithm for this classification is given below:

1. For each transaction to be classified, read one by one the decision rule from the Decision table.
2. Match the fields from the transaction with each decision rule. (Fields having blank entries in decision table indicate don't care condition).
3. First try to find out perfect match & fill the Class field of the transaction with the class of matched rule.
4. If perfect match doesn't find then find the best match where best match is found based on maximum match count & fill the Class field of the transaction with the class of best matched rule.

IV. RESULTS

A. Data Sets

Around 1 lac credit card transactions are generated based on the different fraud situations. Out of it transactions are divided into two major sets of 50780 transactions. The detailed summary of dataset is given in Table I.

TABLE I
SPECIFICATION OF EACH TRANSACTION SET

Test Set Name	No. of Transactions
Test1	10000
Test2	20000
Test3	30000
Test4	40000
Test5	10000
Test6	20000
Test7	30000
Test8	40000
Set 1	50780
Set 2	50780
Main Set	101560

The classifier is trained with different transaction sets & used for the classification of each of these sets. For comparison purpose basic ID3 algorithm & ID3 with pre-pruning named as PruneID3 algorithm is used for training. As classes of these transactions are already known the classification accuracy is evaluated by comparing the classified transactions with the original class value of the transactions. Classification measures used for results evaluation are True Positive Rate (TPR), False Positive Rate (FPR), and Accuracy [7].

B. Local and Agent Based Classification Results

From around 1 lac transactions some transactions are taken for training & part of it are taken for testing purpose & then this procedure is repeated for the whole transaction database.
In case of agent based classification the decision rules formed using the transactions of remote database are imported to classify the local transaction sets. So these results are evaluated by classifying one test file with decision rules formed by different train files.
The results evaluated with both the training algorithms (ID3 & PruneID3) are listed & compared.

a) Comparison of Accuracy & Classification measures with Main Set as training file is shown:

TABLE II
COMPARISON OF TRUE POSITIVE RATE (TPR)

	True Positive Rate (TPR) in %	
Test Sets	ID3	PruneID3
Test1	84.67	87.53
Test2	84.27	87.74
Test3	79.91	89.03
Test4	81.44	88.67
Test5	84.78	83.56
Test6	83.92	87.83
Test7	79.91	89.11
Test8	81.33	88.76
Set1	81.88	88.47
Set2	82.35	84.45

Table II shows that PruneID3 gives average 87% of TPR whereas ID3 gives 82% of TPR.

Table III gives comparative results of False Positive Rate with ID3 & PruneID3 algorithm.
Results show that PruneID3 gives lower FPR around average 12% than ID3 which gives average FPR of 30%.

TABLE III
COMPARISON OF FALSE POSITIVE RATE

	False Positive Rate (FPR) in %	
Test Sets	ID3	PruneID3
Test1	33.52	12.99
Test2	33.45	12.38
Test3	31.6	15.93
Test4	32.34	15.24
Test5	23	9.84
Test6	31.59	8.09
Test7	29.32	12.32
Test8	30.1	11.44
Set1	32.55	14.62
Set2	22.43	12.06

Table IV represents results of overall accuracy evaluation. Overall accuracy of each transaction getting classified to the correct class level is very important for fulfilling the objective of the system. Considering the main set as base classifier & classifying different data sets PruneID3 gives highest average accuracy of 80% which is much better than ID3 giving overall accuracy of 62%.

TABLE IV
COMPARISON OF OVERALL ACCURACY

	Accuracy in %	
Test Sets	**ID3**	**PruneID3**
Test1	60.82	83.09
Test2	61.21	83.56
Test3	59.07	83.63
Test4	59.02	83.49
Test5	68.25	69.22
Test6	65.83	73.55
Test7	59.86	83.47
Test8	60.22	83.09
Set1	59.42	83.55
Set2	62.58	79.69

b) Level wise comparison of accuracy with training file (main set) consisting of around 1 lac transactions

Table 5 depicts results of evaluated accuracy for class type L1 with ID3 & PruneID3 algorithm.

TABLE V
ACCURACY EVALUATION OF CLASS TYPE L1

	Accuracy of L1 in %	
Test Sets	**ID3**	**PruneID3**
Test1	66.48	87.01
Test2	66.55	87.62
Test3	68.4	84.07
Test4	67.66	84.76
Test5	77	90.16
Test6	68.41	91.91
Test7	70.68	87.68
Test8	69.9	88.56
Set1	67.45	85.38
Set2	77.57	87.94

Table VI depicts results of evaluated accuracy for class type L2 with ID3 & PruneID3 algorithm.

TABLE VI
ACCURACY EVALUATION OF CLASS TYPE L2

	Accuracy of L2 in %	
Test Sets	**ID3**	**PruneID3**
Test1	60.62	81.81
Test2	60.82	82.47
Test3	59.55	79.61
Test4	59.38	80.1
Test5	60.69	84.27
Test6	61.01	84.22
Test7	60.29	81.44
Test8	59.95	81.96
Set1	59.96	80.60
Set2	59.92	84.03

Table VII shows results of evaluated accuracy for class type L3 with ID3 & PruneID3 algorithm.

TABLE VII
COMPARISON OF ACCURACY OF CLASS L3

	Accuracy of L3 in %	
Test Sets	**ID3**	**PruneID3**
Test1	77.97	97.04
Test2	78.05	97.09
Test3	64.25	95.26
Test4	68.48	95.81
Test5	70.15	59.06
Test6	70.22	60.36
Test7	61.64	85.02
Test8	65.01	85.09
Set1	70.32	96.03
Set2	65.85	85.25

TABLE VIII
COMPARISON OF ACCURACY OF CLASS L4

	Accuracy of L4 in %	
Test Sets	**ID3**	**PruneID3**
Test1	43.93	70.98
Test2	44.26	70.84
Test3	41.6	79.76
Test4	40.97	76.95
Test5	62.9	43.19
Test6	60.04	52.02
Test7	42.93	78.56
Test8	43.91	75.49
Set1	41.22	75.79
Set2	47.82	63.82

Table VIII gives results of evaluated accuracy for class type L4 with ID3 & PruneID3 algorithm.

Observations show that PrueID3 performs well in all the comparisons than ID3 algorithm. Also accuracy of each class type with PruneID3 is better than ID3.

C. Results of meta-classification

Meta-classification is learning from learned knowledge. Here the system applies class combiner policy [1] where the decision rules generated by different base classifiers are combined & then applied to classify the transactions. The rule sets are formed with combination of different decision rules generated using different training files.

The decision rule sets are formed with combination of different decision rules generated using different training files. Description of these sets is given below & results are evaluated with the same.

Rset1 has decision rules formed by combining rules generated by using Test1, Test2, Test3, Test4 as training file and PruneID3 as learning algorithm.

Rset2 contain decision rules formed by combining rules generated by using Test1, Test2, Test3, Test4 as training file and ID3 as learning algorithm.

Rset3 has decision rules formed by combining rules generated by using Test1, Test2 with ID3 algorithm and Test3, Test4 with PruneID3 as learning algorithm.
Rset4 comprised of decision rules formed by combining rules generated by using Test1, Test2 with PruneID3 algorithm and Test3, Test4 with ID3 as learning algorithm.
Data set **Dset1** contains 30000 transactions formed by combining transactions from the data sets Test1 & Test2.

TABLE IX
META-CLASSIFICATION RESULTS OF DSET1

	File used to form Decision Table			
	Rset1	**Rset2**	**Rset3**	**Rset4**
L1	90.91	45.37	92.75	83.72
L2	85.22	47.58	75	57.8
L3	91.18	64.45	87.46	70.34
L4	78.54	77.59	83.67	63.13
FPR	9.09	54.63	7.25	16.28
FNR	12.82	6.32	7.72	7.7
TPR	87.18	93.68	92.28	92.3
Accuracy	86.19	58.36	85.07	69.43

Results of Table IX are one of the samples of meta-classification. Observation of meta- classification results show that classification accuracy of those classifiers is high which are formed with the rules of PruneID3 algorithm. Also, their FPR is low but TPR with the classifiers formed with ID3 algorithm is higher than the others. This reflects the advantage of meta-classification where benefits of two different algorithms can be combined to get better results.

V.CONCLUSION

ID3 is a basic decision tree classification algorithm. Credit card fraud detection system is one of the applications of it which has been developed. The application is useful for inter-banking where banks can share their fraud detecting rules with each other to overcome the threat of fraud which is spreading widely in world of credit cards. The system developed follows homogeneous approach i.e. the same algorithm is used for local & meta-classification.

In contrast to previously developed credit card fraud detection systems where transactions were getting classified in only two levels either fraud or non-fraud the system developed can differentiate among different fraudulent situations & classifier transactions in four levels where level wise fraud risk increases.

The performance based on Accuracy & True Positive Rate is compared between simple ID3 algorithm and modified ID3 algorithm named as PruneID3.

Around 101580 transactions were generated & divided into sets of 10000, 20000, 30000, 40000 & 50780 transactions. The classifier was then trained with these different sets & accuracy was evaluated by classifying these sets with these different decision trees.

PruneID3 gives on an average 80% accuracy whereas Simple ID3 algorithm gives on an average 62% accuracy. Fraud catching rate (TPR) of both the classifiers is 85%. False Alarm rate (FPR) of PruneID3 is 12% and ID3 gives False alarm rate of 30%.

PruneID3 algorithm is decision tree learning algorithm with pruning. Observation shows that at level 4 it gives highest accuracy for different transaction sets of the application. But then also there is always an 'optimum' pruning level for different applications & requirements that one has to identify and select.
The classification does not consider the customer ID & thus it gives customer independent classification.

Scalability is one of the features provided by the system where database is spread across the network & only decision table (small in size) is used to classify them. Meta-classification provides the facility of combining the learned rules based on the different transaction sets & applies it on different databases for classification.

Number of decision rules generated by PruneID3 is much lesser than rules generated by ID3 algorithm. Many times rules generated by ID3 are redundant & meaningless.

In case of PruneID3 algorithm, rules are lesser which directly affects the size of the decision table & also time required to perform classification of large number of transactions.

As size of the decision table generated with PruneID3 algorithm is small the required network bandwidth while transferring the decision table through agents also reduces.

VI.REFERENCES

[1] Salvatore, Philip et al:, "Meta learning agents for fraud and intrusion detection in Financial Information Systems.", *Inv paper Proceedings in International conference of Knowledge Discovery and Data mining, 1996.*

[2] S. Stolfo et al., "JAM: Java Agents for Metalearning over Distributed Databases," *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining*, AAAI Press, *Menlo Park,Calif., 1997, pp. 74–81.*

[3] Philip K. Chan, Wei Fan, Andreas L. Prodromidis, and Salvatore J. Stolfo, "Distributed Data Mining in Credit Card Fraud Detection", *Proc.IEEE Int'l Conf. Intelligent Systems, Dec.1999.*

[4] Jiawei Han, Micheline Kamber, "Data Mining Concepts and Techniques", *pp. 279-328, 2001.*

[5] Salvatore J. Stolfo, David W. Fan, Wenke Lee and Andreas L. Prodromidis, "Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results", *DARPA, 1999.*

[6] Zhang Yong, "Decision Tree's Pruning Algorithm Based on Deficient Data Sets", *In Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2005.

[7] Arun Poojari, "Data Mining techniques", pp 150 -200, 1999

[8] Szappanos Tibor, Zolotova Iveta, "Distributed Data Mining and Data Warehouse", *ASR '2005 Seminar, Instruments and Control, Ostrava, April 29, 2005.*

[9] J.R. Quinlan, "Induction of Decision Trees, in Machine Learning", 106-181, 1986

[10] Tom. M. Mitchell, "Machine Learning", McGraw-Hill Publications, 1997

VI. BIOGRAPHIES

Shalini Bhatia was born on August 08, 1971. She received the B.E. degree in Computer Engineering from Sri Sant Gajanan Maharaj College of Engineering, Amravati University, Shegaon, Maharashtra, India in 1993, M.E. degree in Computer Engineering from Thadomal Shahani Engineering College, Mumbai, Maharashtra, India in 2003. She has been associated with Thadomal Shahani Engineering College since 1995, where she has worked as Lecturer in Computer Engineering Department from Jan 1995 to Dec 2004 and as Assistant Professor from Dec 2004 to Dec 2005. Since Jan 2006 she is looking after the department as the Head. Her research interests include neural networks, fuzzy systems, bioinformatics, intelligent systems, distributed computing, image processing, and advanced computer architecture. She has published a number of technical papers in National and International Conferences. She is an active member of CSI and also a member of Special Interest Group in Artificial Intelligence (SIGAI) which is a part of CSI.

Dipti Thakur was born on August 20, 1981. She received the B.E. degree in Computer Engineering from Thadomal Shahani Engineering College, Mumbai University, Bandra, Maharashtra, India, in 2002, and pursuing M.E. degree in Computer Engineering from Thadomal Shahani Engineering College, Mumbai, Maharashtra, India. She has been associated with Vidyavardhini's College of Engineering, Vasai, Maharashtra, India, since 2003, where she is working as Lecturer in Computer Engineering Department. Her research interests include Data mining, Data warehousing and Image Processing.