Enhanced Data Mining and Decision Tree Techniques for Network Intrusion Detection System

Nareshkumar D Harale, Dr. B B Mehsram

Abstract -A Network intrusion detection system (IDS) is a security layer to detect ongoing intrusive activities in computer networks and the major problem with IDS is that typically so many alarms are generated as to overwhelm the system operator, many of these being false alarms. Although smart intrusion and detection strategies are used to detect any false alarms within the network critical subnets of network infrastructures, reducing false positives is still a major challenge.

Up to this moment, these strategies focus on either detection or response features, but often lack of having both features together. Without considering those features together, intrusion detection systems probably will not be able to highly detect on low false alarm rates. To offset the above mentioned constraints, this paper proposes a technique to emphasis on detection involving statistical analysis of both attack and normal traffics based on the training data set of KDD Cup 99. This technique also includes a hybrid statistical approach which uses Data Mining and Decision Tree Classification which results reduction misclassification of false positives and distinguish between real attacks and false positives for the data of KDD Cup 99.

Since this technique can be used to evaluate and enhance the capability of the IDS to detect and at the same time to respond to the threats and benign traffic in critical network subnets, application and database infrastructures.

Keywords: Intrusion Alert, False Positive, False Negative, Intrusion Detection System, Data Mining, Decision Tree Classification, Network Subnets.

1. INTRODUCTION

Since June 2001 until November 2001, workstation communities around the world including Malaysia have been trapped in the biggest computer infrastructures attacks in the Internet technology history.

The statistical attacks reported by the Malaysian Computer Emergency Response Team (MyCERT) shows that 17,829 computers within that period had been infected by Nimda and Code Red attacks. The cost to recover all damages caused by these attacks was estimated at about RM22 million [6]. The amount was not inclusive of cost for lost business opportunities due to these attacks. MyCERT argues that several precautions needs to be taken in order to prevent Viruses and other security threats infecting computers, which in turn can help to minimize the cost of recovery.

One possible precaution is the use of an Intrusion Detection System (IDS). IDS are an effective security technology, which can detect, prevent and possibly react to the attack [10]. It monitors target sources of activities, such as audit and network traffic data in computer or network systems, which deploys various techniques in order to provide security services. Therefore, the main objective of IDS is to detect all intrusions in an efficient manner [2]. For example, this may lead to an earlier detection of viruses and worms, and an early warning system in case of a computer virus outbreak. Moreover, the effectiveness of IDS also needs to distinguish between incidents and "normal" alerts. This implies that while the number of false alarms should be reduced, real attacks should not go unnoticed to be effective. Thus, it is important for IDS to be efficient so that the number of false positives and false negatives can be reduced [2].

Statistically, false positive and false negative are always referred as Type I error (i.e. also known as an error, or false positive) and type II error (i.e. also called as ß error, or a false negative). These errors are normally used to describe possible errors made in a statistical decision process [7].

An IDS also acts by labeling alerts as incidents or as non-incidents. In an ideal situation, users may provide feedback by disagreeing or agreeing with the decision made by IDS.

Normally, an input of IDS can be provided by one or more sensors. Multiple sensors can be used as input to a single analyzer and works as observation points on the network. These sensors normally generate many alerts [13]. However, not all of these alerts are relevant because all alerts are analyzed, and only relevant alerts are reported as incidents. The overview of this process is depicted in Fig.1. The input for this process, consisting of alerts, is provided by multiple sensors.

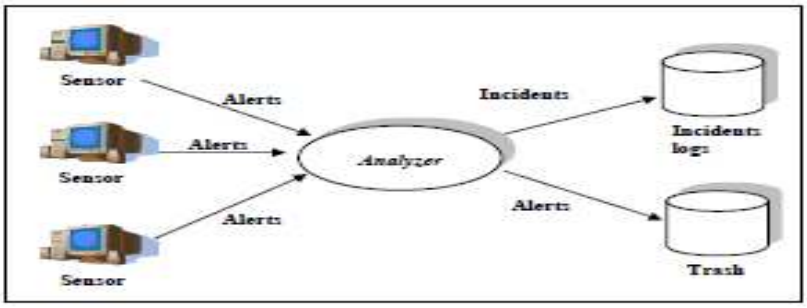


Figure 1: Alerts analyzing that was generated by distributed sensors.

In order to have a real intrusion alarm, all activities needs to be analyzed by the analyzer (Fig. 1). The efficiency of the result from the analyzing process can be increased by using artificial intelligence and machine learning techniques [9] mentions that some tasks cannot be easily defined properly by human expert compared to the effectiveness of a computer generating analysis because it will be difficult for human beings to find relationships and correlations in vast amounts of data [9].

During this research, an attempt is made to filter incidents from alerts. The filtering process is needed to pre-identify the real intrusion activities. This classification is done by using data mining and decision tree techniques.

2. NETWORK INTRUSION DETECTION SYSTEMS USING DATA MINING AND DECISION TREE

Detection method in IDS can be divided into two categories: anomaly detection and misuse detection categories. The anomaly detection strategy looks for unusual and abnormal activities in a system or network, whereas the misuse detection strategy searches for defined or pre-rules done manually by human. Anomaly detection systems regulate normal user behavior profiles and also recognize intrusions by detecting some discrepancy from the normal behavior. Although anomaly detection is able to detect previously unknown security attacks occasionally, it requires huge amount of data to be observed to produce user behavior profiles. Furthermore, anomaly detection causes rather high false alarm rates because any new user behavior which is not included in the user behavior profile is considered an intrusion [10].

Meanwhile, misuse detection can spot intrusion by matching security activities against predefined security attack patterns, which are stored in a database of previously known attacks. Misuse detection methods are usually able to recognise attacks with very high certainty, which is applied in a number of commercial IDS. However, misuse detection cannot identify novel or new intrusions because their pattern is not defined and stored in a database unless it necessitates updating the database and the software system whenever new types of security attacks are discovered [10].

The problem of identifying novel intrusion through misuse detection can be solved statistically in this project. We propose the use of artificial intelligence technique such as data mining and decision tree. Data mining is the best option because previously well known attacks are stored in a database. A mining algorithm such as decision tree is used to analyse previously known attacks to generate a classifier for attacks. The accuracy of the algorithms is measured by

the percentage of false positive and false negative that was generated during the classifying process. A higher of false positive means a lower accuracy and precision of the classifier. A higher false negative implies that the recall of the classifier is lower.

Applying data mining with a decision tree for the development of IDS provides some advantages compared to the classical approach because decision tree gains more quantity of valuable information which in turn can help to enhance the decision on identifying the attacks. While IDS which utilizes crisp values may loose a large amount of valuable information, the decision tree provides some flexibility to the uncertain problem of intrusion detection, thus allowing a much greater complexity for IDS. We performed experiments to classify the network traffic patterns according to the basic 5-class taxonomy, also based on the 23-attack-instance taxonomy. The five classes of patterns in the DARPA data are discussed in the next section. According to [4], it is shown that using a decision tree for classification gives a high accuracy which in turn can help to reduce training and testing times compared to the traditional neural network.

3. DATA STRUCTURES FOR INTRUSION DETECTION

In 1998, under DARPA intrusion detection evaluation programme, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical US. Air Force LAN. The LAN was operated like a real environment, but was blasted with multiple attacks [5].

For each TCP/IP connection, 41 various quantitative and qualitative features were extracted [11]. Of this database, a subset of 494021 data were used which compromised 20% of normal patterns. Attack types were divided into the following 4 main categories:

3.1 Probing Attack

Probing is a class of attacks where an attacker scans a network to gather information in order to find known vulnerabilities. An attacker with a map of machines and services that are available on a network can manipulate the information to look for exploits. There are different types of probes: some of them abuse the computer's legitimate features; and some of them use social engineering techniques. This class of attacks is the most common because it requires very little technical expertise.

3.2 Denial of Service Attack

Denial of Service (DOS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate

requests, denying legitimate users access to a machine. There are different ways to launch a DOS attack: by abusing the computers legitimate features; by targeting the implementations bugs; or by exploiting the system's misconfigurations. DOS attacks are classified based on the services that an attacker renders unavailable to legitimate users.

3.3 User to root Attack

In this attack, an attacker starts with access to a normal user account on the system by gaining root access. Regular programming mistakes and environment assumption give an attacker an opportunity to exploit the vulnerability of root access. An example of this class of attacks is regular buffer overflows.

3.4 Remote to User Attack

This attack happens when an attacker sends packets to a machine over a network that exploits the machine's vulnerability to gain local access as a user illegally. There are different types of R2U attacks; the most common attack in this class is done by using social engineering. The solution to classify this type of attack was done by many researchers with different approaches and techniques. Most of them use artificial intelligence approaches such as neural network, fuzzy logic, Bayesian, genetic Identifying False Alarm For Network Intrusion Detection System Using Hybrid Data Mining And Decision Tree pp.101-115 algorithms and SVM [1],[4]. In this project, we applied the decision tree technique based on C5.0 algorithms to classify attack. The results of this classification are presented by false positive and false negative numbers.

4. EXPERIMENTAL RESULT

We use data mining software tools with decision tree algorithms known as See5/C5.0 version 2.04. The software is available for demo and evaluation provided by RuleQuest [12], See5/C5.0 is a GUI based software and easy to use. See5/C5.0 is capable of classifying large volumes of data within a second depending on the speed and specification of computer processor. See5/C5.0 classifier provides two data mining algorithms: decision tree algorithm and rule-based algorithm. The authors conducted the experiments in two batches.

The first experiment used a decision tree classifier onto the 10% of KDD Cup 99 training dataset. The second part of this experiment used an algorithm of C5.0's rule-based classifier in order to compare the accuracy result with decision tree. 10% of KDD Cup 99 training dataset was used due to the limitation of machine specification previously used to perform this classification. Moreover, most of IDS classification experiment was done using this dataset [1].

4.1 Decision Tree

Data mining See5 ran very well for the decision tree classifier. As a result, less than 2 minutes was taken to train knowledge from 10% of dataset which contained about 494,021 network traffic records. See5/C5.0 constructs a decision tree from the 494,021 training cases in the file of 'kddcup.data'. The sample of tree structure for decision tree is shown in Fig. 2. The output in Figure 2 is very difficult to comprehend. Although it may not look much like a tree, this output can be paraphrased as IF-THEN statements.

```
See5 [Release 2.04]

Read 494021 cases (41 attributes) from kddcup.data
Decision tree:

wrong_fragment > 0:
:...src_bytes <= 754: teardrop (979)
:   src_bytes > 754: pod (259)
wrong_fragment <= 0:
:...land = 1: land (22/1)
    land = 0:
    :...same_srv_rate <= 0.32:
        :...dst_host_diff_srv_rate <= 0.14:
        :   :...src_bytes <= 0: neptune (106235/24)
        :   :   src_bytes > 0:
        :   :   :...dst_host_count <= 11: buffer_overflow (3/1)
        :   :       dst_host_count > 11:
        :   :       :...src_bytes <= 6: satan (16/2)
        :   :           src_bytes > 6: normal (178/2)
        :   dst_host_diff_srv_rate > 0.14:
        :   :...src_bytes > 19: normal (256/1)
        :       src_bytes <= 19:
        :       :...dst_host_count <= 96: neptune (17)
        :           dst_host_count > 96:

        :                       dst_host_same_src_port_rate > 0.57:
        :                       :...src_bytes <= 0: portsweep (4)
        :                           src_bytes > 0: satan (53)
```

Figure 2: Excerpt of tree structure ased on the result showed in Table 1,

The number and percentage of cases misclassified were 131 or 131/494021 * 100% = 0.03%. The accuracy of C5.0 was measured by the number of false positives and false negatives produced. The value of cases in Table 1 and Table 2 is a number of data classes that represents a 10% dataset. The value for this number is given by DARPA.

Table 1: The Total of False Positive and False Negative for the Class of Attack using Decision Tree

Class	Cases	False Positive	False Negative
Normal	97278	75	15
DoS	391458	9	19
Probe	4107	9	55
R2L	1117	6	23
U2R	59	32	19
Total	**494021**	**131**	**131**

Meanwhile, the value of false positive and false negative is the value generated by C5.0 to represent the classification accuracy. The result shows that only 75 cases were false positive with the ratio of 75/97278 * 100% = 0.08% or 99.92% statistical significant accurate. The 75 cases imply the failure of the classifier to classify the attacks as normal, while only 15 cases of normal record were classified as attacks. The C5.0 was significantly accurate when classifying the DoS attacks with the ratio of 99.99%, Probe is 99.78%, and R2L is 99.46%. However, the accuracy ratio for U2R attack is very low with 45.76% not accurate. This implies that data mining and the decision tree using C5.0 algorithms are not suitable for classifying the U2R attack because the number of records in training dataset was very small. There are only 59 (0.01%) cases representing U2R.

Table 2: shows the detail of the total number of false positive and false negative for types of attack in the decision tree. A total of 131 attacks are classified as false positive and negative. The zero number of false positive and false negative shows that the experiment was significantly accurate in classifying the phf and teardrop attacks. Smurf attack has the biggest number of cases recorded and C5.0 is able to classify 99.99% of the record as a Smurf attack. Phf is an attack type with the smallest number of cases recorded and 5.0 was very successful in classifying phf. However, Spy attack is the smallest number recorded and is significantly misclassified as different types of attack record, meaning that, regardless of whether the number of cases is the smallest or biggest, the classification is not dependent on the number of cases. On the contrary, it depends on the value of 41 attributes represented. The values of the attributes are very similar to each other especially to a normal record which is misclassified as Satan (12), warezclient (10), back (9) and etc [3].

From the 41 attributes of record KDD Cup '99, only 20 of the attributes were used for C5.0 decision trees classifier. By using only 20 attributes, C5.0 is able to classify the type of attack. The ratios of the attribute usage are shown in Table 3. Some machine learning technique such as neural network, fuzzy logic and support vector machine (SVM) are dependent on the input attributes. The attribute usage obtained can also be applied to neural network, fuzzy logic and SVM [1].

Table 2: The Total of False Positive and False Negative for Types of Attack Decision Tree

Attacks	Cases	False Positive	False Negative
Back	2203	1	9
buffer_overflow	30	4	2
ftp_write	8	0	5
guess_passwd	53	0	2
Imap	12	0	2
Ipsweep	1247	3	3
Land	21	1	0
loadmodule	9	0	7
Multihop	7	1	3
Neptune	107201	28	1
Nmap	231	1	10
Normal	97278	75	15
Perl	3	1	0
Phf	4	0	0
Pod	264	0	5
Portsweep	1040	1	27
Rootkit	10	0	9
Satan	1589	4	15
Smurf	280790	6	2
Spy	2	0	2
Teardrop	979	0	0
warezclient	1020	3	10
warezmaster	20	2	2
Total	**494021**	**131**	**131**

Table 3: Attribute usage for decision tree classifier

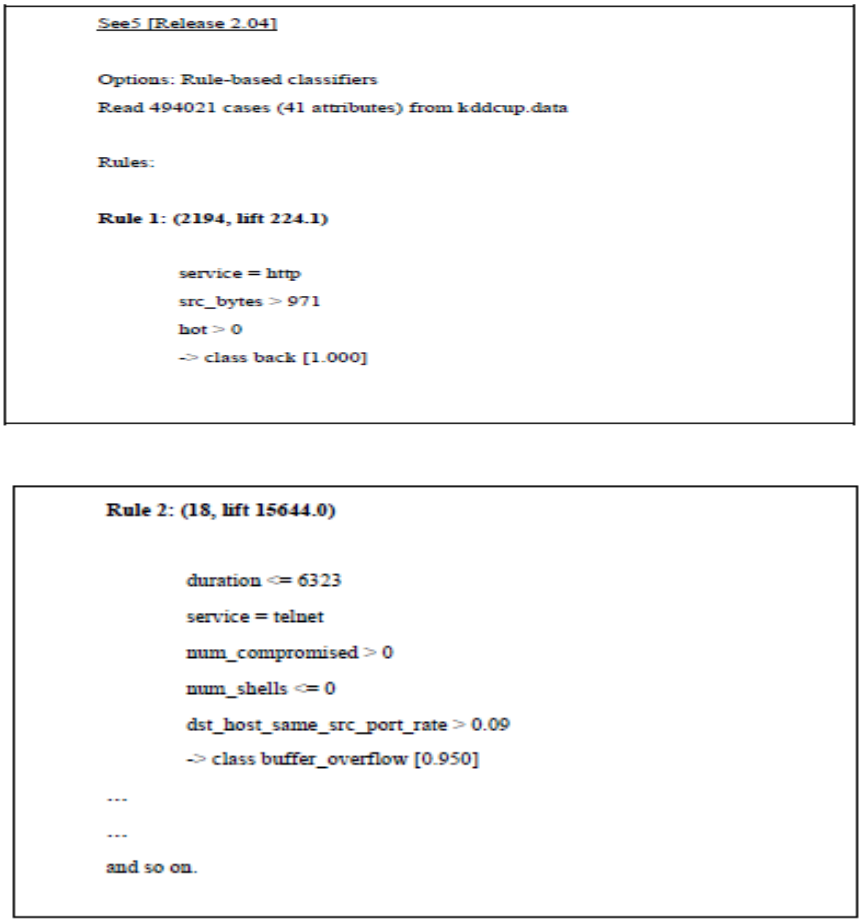
Attributes	Percentage
wrong_fragment	100%
land	100%
same_srv_rate	100%
dst_host_diff_srv_rate	99%
src_bytes	83%
dst_host_serror_rate	78%
num_compromised	78%
num_failed_logins	77%
dst_host_srv_diff_host_rate	77%
hot	77%

root_shell	77%
dst_host_same_src_port_rate	77%
duration	77%
srv_serror_rate	77%
protocol_type	57%
dst_host_srv_count	19%
dst_bytes	18%
count	18%
logged_in	2%
dst_host_srv_rerror_rate	1%

4.2 Rule-based Classifiers

Decision trees can sometimes be quite difficult to comprehend when the tree size is too big. To offset the drawback of decision trees, See5 can generate classifiers called rulesets that consists of unordered collections of (relatively) simple if-then rules. Therefore, the second experiment applied the rule-based algorithms to classify the KDD dataset. The time used for the classification process is less than 2 minutes, as same as the decision tree. Rulesets are generally easier to understand compared to the trees since each rule describes a specific context associated with a class or an attribute. Plus, ruleset is easier to understand than decision tree.

For each rule, the number showed in brackets implies the records that were matched with the rules. For example, Rule 1, (2,194, lift 224.1) means that 2,194 number of 49,4021 records was matched with these rules and lift 224.1 is the result of dividing the rule's estimated accuracy by the relative frequency of the predicted class in the training set (see Figure 3. This output also can be paraphrased as IF-THEN statements.



See5 [Release 2.04]

Options: Rule-based classifiers
Read 494021 cases (41 attributes) from kddcup.data

Rules:

Rule 1: (2194, lift 224.1)

service = http
src_bytes > 971
hot > 0
-> class back [1.000]

Rule 2: (18, lift 15644.0)

duration <= 6323
service = telnet
num_compromised > 0
num_shells <= 0
dst_host_same_src_port_rate > 0.09
-> class buffer_overflow [0.950]

---
and so on.

Figure 3: Excerpt of Rule sets Constructed by C5.0

The result in Table 4 shows that 128 attack records were classified as normal (false positive). This number is bigger than the result from Table 2 which means that rule-based classification generates a higher number of false positive than decision tree. In the next case, Table 5 shows that C5.0 accuracy in classifying the DoS attacks was 99.99%, Probe was 99.85%, and R2L was 99.82% and 94.92% respectively.

Table 4: The Total of False Positive and False Negative for Rule-based classifier

Attacks	Cases	False Positive	False Negative
Back	2203	0	9
buffer_overflow	30	1	8
ftp_write	8	0	6
guess_passwd	53	0	1
Imap	12	0	2
ipsweep	1247	2	3
Land	21	1	0
loadmodule	9	0	7
multihop	7	1	3
neptune	107201	26	4
Nmap	231	0	27
Normal	97278	128	12
Perl	3	1	0
Phf	4	0	4
Pod	264	0	5
portsweep	1040	2	24
Rootkit	10	0	10
Satan	1589	2	20
Smurf	280790	21	2
Spy	2	0	2
teardrop	979	0	0
warezclient	1020	2	18
warezmaster	20	0	20
	494021	**187**	**187**

Table 5: The Total of False Positive and False Negative for the Class of Attack using Rule-based Classifier

Class	Cases	False Positive	False Negative
Normal	97278	128	12
DoS	391458	48	20
Probe	4107	6	74
R2L	1117	2	49
U2R	59	3	28
	494021	**187**	**187**

5. DISCUSSION

The number of observations and conclusion are drawn from the results illustrated in Table 6. It shows that the performance comparisons of false alarm rate using decision tree provides a more accurate classification of Normal, DoS and R2L than the rule-based classifier. However, the rule-based classifier was more accurate when classifying class Probe and U2R because it generates lower false alarm rate.

Table 6: Performance Comparison of False Alarm Rate for the Class of Attack using Decision tree and Rule-based classifier.

Class	False Alarm Rate for Decision Tree (%)	False Alarm Rate for Rule-based Classifier (%)
Normal	0.015	0.025
DoS	1.822×10^{-3}	9.716×10^{-3}
Probe	1.822×10^{-3}	1.215×10^{-3}
R2L	1.215×10^{-3}	4.048×10^{-4}
U2R	6.477×10^{-3}	6.073×10^{-4}

The accuracy of decision tree in classifying the normal record is higher than rule-based classification (Table 6). Since the acceptable levels of false alarms for IDS is less then 10%, the decision tree classification and rule-based classification are suitable for use as an IDS model because the false alarm rate for class normal is 1.5% and 2.5% as shown in Table 6. However, the acceptable levels of false alarm can be higher or lower depending on the level of IDS tuning and the type of traffic on a network [13].

6. CONCLUSION

We have tested and demonstrated the significance of decision tree for modeling of network intrusion detection system for class of normal, DoS, and R2L. For the class of Probe and U2R, rule-based classification approach is more appropriate and convenient. However, based on acceptable levels of false alarm rate, decision tree is more suitable than rule-based for modeling network intrusion detection systems.

7. REFERENCES

1. Ajith Abraham, Ravi. Jain, Soft Computing Models for Network Intrusion Detection Systems. Classification and Clustering for Knowledge Discovery, Saman Halgamuge and Lipo Wang (Eds.), Studies in Computational Intelligence, Vol. 4, Springer Verlag Germany, 2005, ISBN: 3-540-260730 ,Chapter 13, pp. 187-204.

2. Gowadia, V., Farkas, C., and Valtorta, M., Paid: A probabilistic agent-based intrusion detection system. Journal of Computers and Security, 2005.

3. Hasimi Sallehudin, "Pengenalpastian Amaran Palsu Positif Menggunakan Penggalian Data dan Pepohon Keputusan". University of Malaya. 2008.

4. Hettich, S. and Bay, S. D., The UCI KDD Archive Irvine, CA: University of California, Irvine, KDD Cup 1999 Data, 5th International Conference on Knowledge Discovery and Data Mining, 1999.

5. Kendall, K. 1999, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, S.M. Thesis, MIT Department of Electrical Engineering and Computer Science, 1999.

6. Malaysian Computer Emergency Response Team (MyCERT), 2007, Last Malaysian Journal of Computer Science, Vol. 21(2), 2008. Identifying False Alarm for Network Intrusion Detection System Using Hybrid Data Mining and Decision Tree pp.101-115].

7. Moulton, R.T., "Network Security", Datamation, Vol.29, No.7, 1983, pp.121-127.

8. MIT Lincoln Laboratory, DARPA Intrusion Detection Evaluation. [Internet] http://www.ll.mit.edu/IST/ideval

9. Nilsson, N., Introduction to Machine Learning. Stanford University, 1996. [Internet] http://ai.stanford.edu/~nilsson/MLDraftBook/MLBOOK.pdf

10. Rebecca Base and Peter Mell, NIST Special Publication on Intrusion Detection Systems. Infidel, Inc., Sctts Valley, CA and National Institute of Standards and Technology, 2001.

11. Rietta, F., Application layer intrusion detection for sql injection. Proceedings of the 2006 ACM Symposium of Applied Computing (ACMSE-2006).

12. Yu, Z., Tsai, J. J. P., and Weigert, T. 2008. An adaptive automatically tuning intrusion detection system. ACM Trans. Autonom. Adapt. Syst. 3, 3, Article 10 (August 2008), 25 pages.

13. Wenke Lee, Sal Stolfo and Kui Mok, A Data Mining Framework for Building Intrusion Detection Models. Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, 1999.

AUTHORS' PROFILE:

Nareshkumar D Harale

Department of Computer Engineering

MGM'sCET, Navi Mumbai, India.

Dr. B B Mehsram

Department of Computer Technology

VJTI Mumbai, India.