ISSN:2231-0711
www.ijcset.net

Robust Intrusion Detection System using Layered Approach with Conditional Random Fields

Vasavi Bande, U.D.Prasan
Department of Computer Science and Engineering
Aditya Institute of Technology and Management
Tekkali, A.P, India

Abstract- **Network Security is fast becoming an absolute necessity to protect the information contained on computer systems worldwide. The ever changing network use and operation along with the public concern for protection of sensitive information makes implementing an efficient and effective security plan a must. Intrusion detection Systems faces a number of challenges, One component of this approach is that of network intrusion detection systems what we are discussing in this paper uses Snort signature database. An intrusion detection system must reliably detect malicious activities in a network and must perform efficiently to cope with the large amount of network traffic. Making a network intrusion detection system work effectively should not degrade overall network performance so some type of performance measurement plan must be designed and implemented. Intrusion Detection Systems have become very popular in the recent years. This is due to the obvious explosive growth of the Internet and the fact that most us keep and access sensitive data online. In this paper, we have addressed these two issues of accuracy and efficiency using conditional random fields and layered approach. We have demonstrated that high attack detection accuracy can be achieved by using conditional random fields and high efficiency by implementing the layered approach. Finally, we have resolved that this system is robust and is able to handle noisy data without compromising the network performance.**

Keywords **– IDS, NIDS, Conditional random fields, layered based, preprocessor, snort signatures, FAR.**

I. INTRODUCTION

An IDS is a security counter measure. It monitors things looking for signs of intruders. The most important activity of the system is intruder detection mechanism. IDS monitor the packets on a network and compare them with a database of signatures or attributes for known malicious threats. This is similar to the way in which most antivirus systems detect viruses. The important problem is that there is a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to the signature database.

To keep the application up to date, in this paper we have used the snort project signatures database. An intrusion detection system (IDS) is a software and/or hardware designed to detect unwanted attempts at accessing, manipulating or disabling of computer systems mainly through a network. These attempts may take the form of attacks, as examples, by crackers, malware or any disgruntled employees. An IDS cannot directly detect attacks with in properly encrypted traffic. An intrusion detection system is used to detect several types of malicious behaviors that can compromise the security and trust of a computer system. This includes network attacks against vulnerable services[1], data driven attacks on applications, host based attacks such as privilege escalation, unauthorized logins and access to sensitive files, and malware (i.e. virus,worms,trojan horses).

II. VARIOUS INTRUSION DETECTION SYSTEMS (IDS)

IDS inspect all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or to compromise a system. An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network [12]. There are different types of IDS but we follow the signature based IDS. Most intrusion detection systems (IDS) are what are known as signature-based. This means that they operate in much the same way as a virus scanner, by searching for a known identity or a signature for each specific intrusion event. And, while signature-based IDS is very efficient at sniffing out known attacks, like anti-virus software, depend on receiving regular signature updates, to keep in touch with variations in hacker's techniques. In other words, signature-based IDS is only as good as its database of stored signatures.

There are so many types of IDS mechanisms:

- Network intrusion detection system (NIDS)
- Host-based intrusion detection system (HIDS)
- Protocol-based IDS(PIDS)
- Application protocol-based intrusion detection system(APIDS).

As we are dealing with only the network packets we are using NIDS. A network intrusion detection system is a system that also tries to detect malicious activity such as denial of service attacks[3], port scans or even attempts to crack into computers by monitoring network traffic[11]. The NIDS does this by reading all the incoming packets and try to find out suspicious patterns. If, for example, a large number of TCP connection requests to a very large number of different ports are observed, one could assume that there is someone conducting a port scan of some or all of the

computer(s) in the network. It also tries to detect incoming shell codes in the same manner that an ordinary intrusion detection systems does. NIDS is not limited to inspecting incoming network traffic only. Often valuable information about an ongoing intrusion can be learned from outgoing or local traffic as well. Some attacks might even be staged from the inside of the monitored network or network segment, and are therefore not regarded as incoming traffic at all. Often, network intrusion detection systems work with other systems as well. They can for example update some firewalls blacklist with the IP addresses of computers used by (suspected) crackers. Certain DISA documentation, such as the Network STIG, uses the term NID to distinguish an internal IDS instance from its outward-facing counterpart.

In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewall's simplistic filtering rules[3]. In a host-based system, the IDS examines at the activity on each individual computer or host.

A *signature based* IDS would monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware as usual practice. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During that lag time your IDS would be unable to detect the new threat.

III. **RELATED STUDY**

The field of intrusion detection and network security has been around since late 1980s. Since then, a number of methods and frameworks have been proposed and many systems have been built to detect intrusions. Various techniques such as association rules, clustering, naive Baye's classifier, Decision trees, support vector machines, neural networks and others have been applied to detect intrusions. In this section, we briefly discuss these techniques and frameworks.

A. Association rule mining

These are based on building classifiers by discovering Relevant patterns of program and user behavior. Association rules are used to learn the record patterns that describe user behavior. These methods can deal with symbolic data, and the features can be defined in the form of packet and connection details. However, mining of features is limited to entry level of the packet and requires the number of records to be large and sparsely populated. Otherwise, they tend to produce a large number of rules that increase the complexity of the system.

B. Data Clustering Methods k-means and the fuzzy c-means

Clustering technique is based on calculating numeric distance between the observations, and hence, the observations must be numeric. Observations with symbolic features cannot be easily used for the clustering methods. It considers the features independently and is unable to capture the relationship between different features of a single record, which further degrades attack detection accuracy.

C. Naive Baye's classifiers

These make strict independence assumption between the attributes in an observation resulting in lower attack detection accuracy when the features are correlated, which is often the case for intrusion detection. Bayesian network can also be used for intrusion detection. However, they tend to be attack specific and build a decision network based on special characteristics of individual attacks. Thus, the size of a Bayesian network increases rapidly as the number of features and the type of attacks modeled by a Bayesian network increases [9]. To detect anomalous traces of system calls in privileged processes, hidden Markov models (HMMs) have been applied in and However, modeling the system calls alone may not always provide accurate classification as in such cases various connection level features are ignored. Further, HMMs are generative systems and fail to model long-range dependencies between the observations.

D. Decision trees

This method selects the finest features for each decision node during the construction of the tree based on some well-defined criteria[8]. One such criterion is to use the information gain ratio. Decision trees generally have very high speed of operation and high attack detection accuracy.

E. Neural Networks

According to Debar though the neural networks can work effectively with noisy data, they require large amount of data for training and it is often hard to select the best possible architecture for a neural network.

F. Support Vector Machines

Support vector machines have also been used for detecting intrusions. Support vector machines map real valued input feature vector to a higher dimensional feature space through nonlinear mapping[6].This can also provide real-time detection capability, deal with large dimensionality of data, and can be used for binary-class as well as multiclass classification. Experimental results on the KDD '99 intrusion data set show that our proposed system based on Layered Conditional Random Fields outperforms other well-known methods such as the decision trees and the naive Baye's. The improvement in attack detection accuracy is very high, particularly, for the U2R attacks (34.8 percent improvement) and the R2L attacks (34.5 percent improvement). Statistical Tests also demonstrate higher confidence in detection accuracy for our method. Other approaches for detecting intrusion include the use of autonomous and probabilistic agents for intrusion detection.

These methods are generally aimed at developing a distributed intrusion detection system. To overcome the weakness of a single intrusion detection system, a number of frameworks have been proposed, which describe the collaborative use of network-based and host based systems, systems that employ both signature based and behavior-based

techniques. The data analyzed by the intrusion detection system for classification often has a number of features that are highly correlated and complex relationships exist between them. When classifying network connections as either normal or as attack, a system may consider features such as "logged in" and "number of file creations" [2]. When these features are analyzed individually, they do not provide any information that can assist in detecting attacks. However, when these features are analyzed together, they can provide meaningful information, which can be helpful for the classification task[15].

IV. LAYERED APPROACH

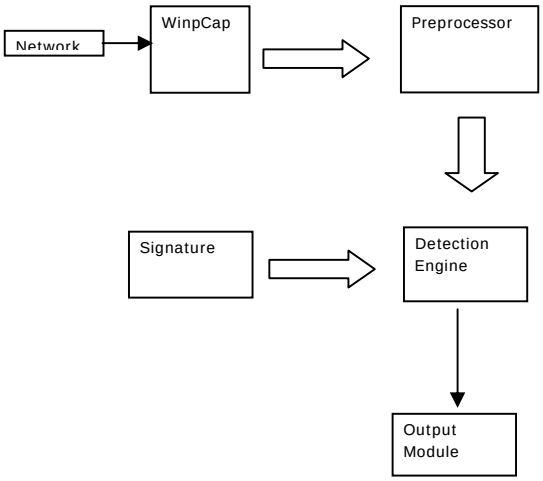

Figure 1. Intrusion Detection System

A. WinPcap

As shown in the figure1, the WinPcap software provides facilities to capture raw packets, both the ones destined to the machine where it's running and the ones exchanged by other hosts (on shared media), filter the packets according to user specified rules before dispatching them to the application, transmit raw packets to the network and to gather statistical values on the network traffic.

B. Preprocessor

As mentioned in the figure1 the preprocessor defines one class called packet and this class will store all the packets that are generated by the WinPcap. It captures all the packets in the Network Interface by using Jpcap captor.

C. Signature Database

It is a specially prepared pattern database. Every incident is analyzed to get a regular expression describing the type of attack attempt. There is lot of signatures in the database for such analysis. We have used signatures of the project Snort because the database is still being developed by the Snort Project Team, so updates are often released. Snort uses a simple [13], lightweight rules description language that is flexible and quite powerful. There are a number of simple guidelines to remember when developing Snort rules. The first is that Snort rules must be completely contained on a single line, the Snort rule parser doesn't know how to handle rules on multiple lines. Snort rules are divided into two logical sections, the rule header and the rule options. The rule header contains the rule's action, protocol, source and destination IP addresses, net masks, source and destination ports information. The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken[14]. Here is an example rule:

```
alert tcp any any -> 192.168.1.0/24 111 (content:"|00
01 86 a5|"; msg: "mountd access";)
```

D. Detection Engine

It takes packets from preprocessor and compares them with special signatures from the database [7]. Result of the compassion is sent to the output module, where a report is prepared. The detection engine compares the packets in the preprocessor and in the signature database. The comparison takes place at different layers. To compare the content in this paper we are using layered approach algorithm, considering different attributes at each layer. Finally it would detect whether the packet has any attack or not.

We now describe the Layer-based Intrusion Detection System (LIDS) in detail. The LIDS draws its motivation from what we call as the Airport Security model, where a number of security checks are performed one after the other in a sequence. Similar to this model, the LIDS represents a sequential Layered Approach and is based on ensuring availability, confidentiality, and integrity of data and (or) services over a network. The goal of using a layered model is to lessen the computations and the overall time required to detect anomalous events. The time required to detect an intrusive event is significant and can be reduced by eliminating the communication overhead among different layers [4]. This can be achieved by making the layers autonomous and self-sufficient to block an attack without the need of a central decision-maker [10].

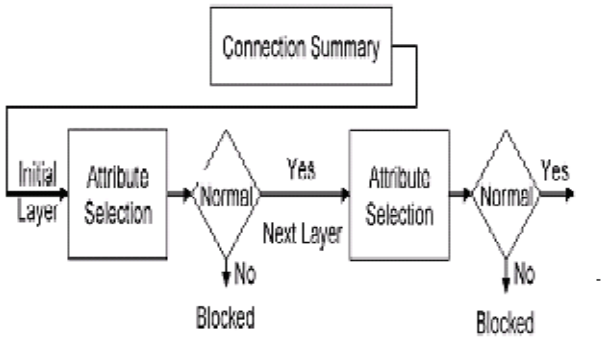

Figure 2. Intrusion detection System at each layer

Every layer in the LIDS framework is trained separately as shown in the figure 2 and then deployed sequentially. We define four layers that correspond to the four attack groups mentioned in the data set. Each layer is then separately

trained with a small set of relevant features. Feature selection is significant for layered approach and discussed is in the section 6. In order to make the layers independent, some features may be present in more than one layer. The layers essentially act as filters that block any anomalous connection, thereby eliminating the need of further processing at subsequent layers enabling quick response to intrusion. The effect of such a sequence of layers is that the anomalous events are identified and blocked as soon as they are detected.

V. CONDITIONAL RANDOM FIELDS

Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting structured data, such as sequences, trees and lattices. The underlying idea is that of defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences. The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference. Additionally, CRFs avoid the label bias problem, a weakness exhibited by maximum entropy Markov models (MEMMs) and other conditional Markov models based on directed graphical models [8]. CRFs outperform both MEMMs and HMMs on a number of real-world tasks in many fields, including bioinformatics, computational linguistics and speech recognition [5]. In sequence modeling, the graph of interest is usually a chain graph. An input sequence of observed variables X represents a sequence of observations and Y represents a hidden (or unknown) state variable that needs to be inferred given the observations. The Y_i are structured to form a chain, with an edge between each Y_{i-1} and Y_i. As well as having a simple interpretation of the Y_i as "labels" for each element in the input sequence, this layout admits efficient algorithms for:

- Model training, learning the conditional distributions between the Y_i and feature functions from some corpus of training data.
- Inference, determining the probability of a given label sequence Y given X.
- Decoding, determining the most likely label sequence Y given X.

The conditional dependency of each Y_i on X is defined through a fixed set of feature functions of the form $f(i,Y_{i-1},Y_i,X)$, which can informally be thought of as measurements on the input sequence that partially determine the likelihood of each possible value for Y_i. The model assigns each feature a numerical weight and combines them to determine the probability of a certain value for Y_i. Hence, the CRFs have proven to be very successful in such tasks, as they do not make any unwarranted assumptions about the data. Hence, we explore the suitability of CRFs for intrusion detection system.

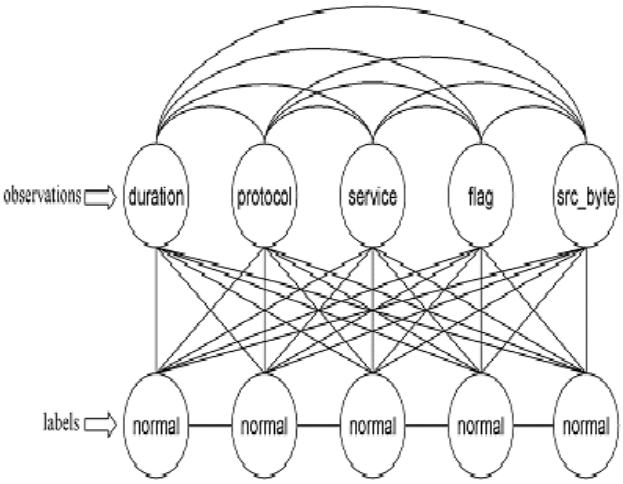

Figure 3 CRF in Pictorial Representation

However, when these features are analyzed together, they can provide meaningful information, which can be helpful for the classification task as shown in the above figure3.

VI. INTEGRATING LAYERED APPROACH WITH CONDITIONAL RANDOM FIELDS

We considered four layers which are mentioned as follows and we have addressed some attributes at each layer and the detailed layer wise intrusion detection system is shown in the figure 4.

A. Probe layer

The probe attacks are aimed at acquiring information about the target network from a source that is often external to the network. Hence, basic connection level features such as the "duration of connection" and "source bytes" are significant while features like "number of files creations" and "number of files accessed" are not expected to provide information for detecting probes.

B. DoS layer

For The DoS layer, we considered traffic features such as the "percentage of connections having same destination host and same service" and packet level features such as the "source bytes" and "percentage of packets with errors" are significant.

C. R2L layer

Detecting the R2L attacks are one of the most difficult as they involve both the network level and the host level features. We have therefore selected both the network level features such as the "duration of connection" and "service requested" and the host level features such as the "number of failed login attempts" among others for detecting R2L attack.

D. U2R layer (User to Root attacks)

The U2R attacks involve the semantic details that are very difficult to capture at an early stage. Such attacks are often content based and target an application. Hence, for U2R attacks, we selected features such as "number of file creations" and "number of shell prompts invoked," while we ignored features such as "protocol" and "source bytes."

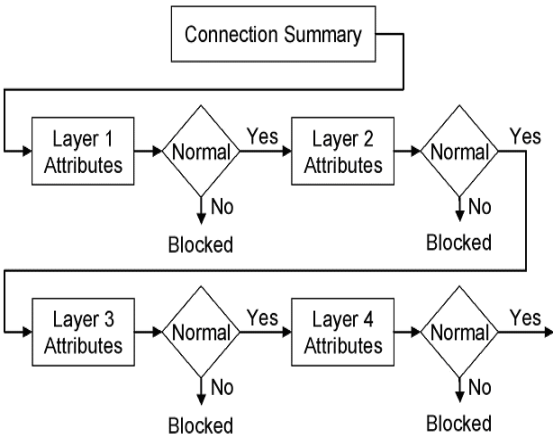

Figure 4 . Detailed Intrusion Detection System at each layer.

VII. IMPLEMENTATION AND RESULTS

As we tried to identify attacks at each layer, the effect of such a sequence of layers is that the anomalous events are identified and blocked as soon as they are detected and as CRFs have proven to be very successful, as they do not make any unwarranted assumptions about the data. Hence, we explore the suitability of CRFs for intrusion detection system. In the following figure 5 we have shown some sample result of the DOS attacks.

A. Algorithm & Training:

The algorithm what we have designed for this paper is as follows, using which we are trying to identify anomalous events and blocked as soon as they are detected.

Step 1: Select the number of layers, n, for the complete system.

Step 2: Separately perform features selection for each layer.

Step 3: Train a separate model with CRFs for each layer using the features selected from Step 2.

Step 4: Plug in the trained models sequentially such that only the connections labeled as normal are passed to the next layer.

Step 5: For each (next) Test Instance perform Steps 6 through 9.

Step 6: Test the instance and label it either as attack or normal.

Step 7: If the instance is labeled as attack, block it and identify it as an attack represented by the layer name at which it is detected and go to Step 5. Else pass the sequence to the next layer.

Step 8: If the current layer is not the last layer in the system, test the instance and go to Step 7. Else go to Step 9.

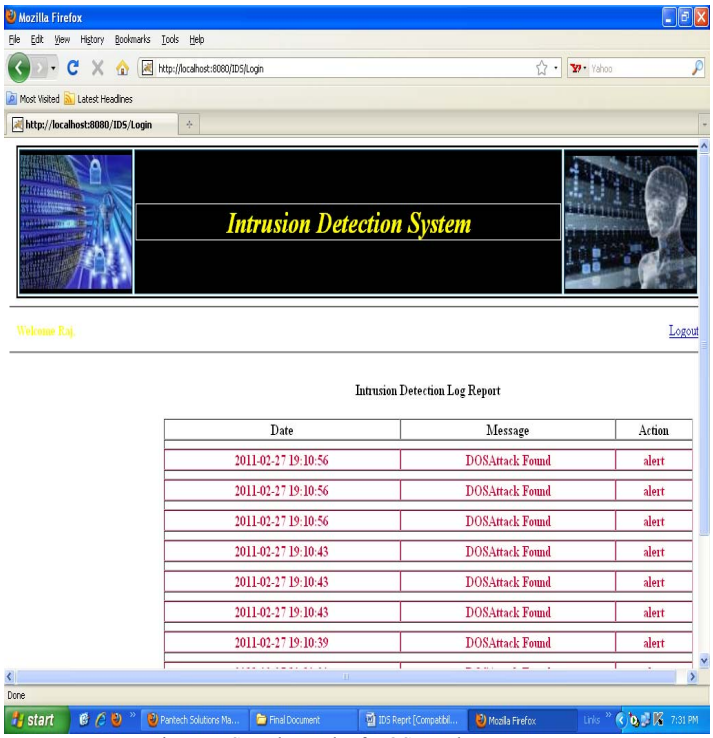

Figure 5 Sample result of DOS attack.

VIII. CONCLUSION

In this paper, we have addressed the dual problem of Accuracy and Efficiency for building robust and efficient intrusion detection systems. Our experimental results are very effective in improving the attack detection rate and decreasing the False Alarm Rating(FAR). Having a low FAR is very important for any intrusion detection system. Further, feature selection and implementing the Layered Approach significantly has reduced the time required to train and test the model. We showed that the sequence labeling methods such as the CRFs can be very effective in detecting attacks. We have compared our approach with some well-known methods and found that most of the present methods for intrusion detection fail to reliably detect R2L and U2R attacks, while our integrated system can effectively and efficiently detect such attacks giving an improvement of 34.5 percent for the R2L and 34.8 percent for the U2R attacks. This approach can help in identifying an attack once it is detected at a particular layer, which expedites the intrusion response mechanism, thus minimizing the impact of an attack and also is robust to noise. Finally, due to layered approach it is helpful to the network administrators. The areas for future research include the use of our method for extracting features that can aid in the development of signatures for much effective signature-based systems. This can further be extended to implement pipelining of layers in multi core processors, which is likely to result in very high performance.

REFERENCES

[1] W. Lee, S. Stolfo, and K. Mok, "Mining Audit Data to Build Intrusion Detection Models," Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98), pp. 66-72, 1998.

[2] Boswell, Wendy (n.d.). *A Short History of the Internet*. http://websearch.about.com/od/whatistheinternet/a/historyinternet.htm.

[3] Chen, Qiang (2001). *Computer Intrusion Detection through Noise Cancellation*. Arizona: Arizona State University.

[4] Cisco (2007). *Understanding Delay in Packet Voice Networks*. http://www.cisco.com/warp/public/788/voip/delay-details.hml

[5] A. McCallum, "Efficiently Inducing Features of Conditional Random Fields," Proc. 19th Ann. Conf. Uncertainty in Artificial Intelligence (UAI '03), pp. 403-410, 2003.

[6] D.S. Kim and J.S. Park, "Network-Based Intrusion Detection with Support Vector Machines," Proc. Information Networking, Networking Technologies for Enhanced Internet Services Int'l Conf. (ICOIN '03), pp. 747-756, 2003.

[7] Inline (2005). *Snort Inline Part II*. Pete Savage. Retrieved on March 8, 2007 http://linuxgazette.net/118/savage.html.

[8] A. McCallum, D. Freitag, and F. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," Proc. 17th Int'l Conf. Machine Learning (ICML '00), pp. 591-598,2000.

[9] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), pp. 420-424, 2004.

[10] NetOptics (n.d.). *White Paper: Deploying Network Taps with Intrusion Detection Systems*. Retrieved on April 4, 2007 http://www.netoptics.com/products/pdf/Taps-and-IDSs.pdf.

[11] Proctor, Paul E. (2001). *The Practical Intrusion Detection Handbook*. New Jersey: Prentice Hall.

[12] Puketza, Nicholas Joseph (2000). *Approaches to Computer Security: Filtering, Testing, and Detection*. California: University of California.

[13] Savage, Pete (2005). *Snort Inline Part I*. Retrieved on March 8, 2007 http://linuxgazette.net/117/savage.html.

[14] Snort (2007). *Snort Users Manual*. Retrieved March 8, 2007 from http://www.snort.org./docs/snort_htmanuals/htmanual_261.

[15] Zamboni, Diego (2001). *Using Internal Sensors for Computer Intrusion Detection*. Purdue: Purdue University

BIOGRAPHIES

Vasavi Bande, is a student of Master of Technology, Computer Science and Engineering in Aditya Institute of Technology and Management affiliated to Jawaharlal Nehru Technological University, Kakinada A.P., India. She has vast experience in Computer Science and Engineering areas pertaining to academics and industry related real time projects. She has presented many papers to her credit at conferences. She has presided over as judge to many Paper Presentations and Technical Quizzes. She has authored 6 research papers to date and are published in reputed and indexed International Computer Science Journals. She has guided 20 Students of Master degree in Computer Science and Engineering in their major projects. She is bestowed with the Editorial Member on three International Journals Boards and is nominated as Reviewer to four International Computer Science and Information Technology Journals. Her area of research include Cloud computing, Network Security, Image Processing, Data Mining, Web Technologies and Emerging Technologies. She can be reached at: vasavi.bande@yahoo.co.in.

U.D.Prasan is B.Tech(CSE), M.Tech(CSE) from JNTU , A.P, India, MISTE, MCSI , Presently working as Associate professor in Aditya Institute of Technology and Management, Tekkali , A.P, India. He has 10 years of experience in teaching Computer Science and Engineering related subjects .He is a research scholar and his area of interest and research include Data mining , Computer Networks and Pattern Recognition and Neural Networks. He can be reached at udprasanna@rediffmail.com.