ISSN 2047-3338

Data Mining Machine Learning Techniques – A Study on Abnormal Anomaly Detection System

M. Sathya Narayana[1], B. V. V. S. Prasad[2], A. Srividhya[3] and K. Pandu Ranga Reddy[4]
[1,2]Visvesvaraya College of Engineering and Technology, Hyderabad, India
[3]SV College of Engineering and Technology, India
[4]Nishitha College of Engineering and Technology, India

***Abstract*– An important research problem in knowledge discovery and data mining is to identify abnormal instances. Finding anomalies in data has important applications in domains such as fraud detection and homeland security. While there are several existing methods to identify anomalies in numerical datasets, there has been little work aimed at discovering abnormal instances in large and complex relational networks whose nodes are richly connected with many different types of links. To address this problem we designed a novel, unsupervised, domain independent framework that utilizes the information provided by different types of links to identify abnormal nodes. Our approach measures the dependencies between nodes and paths in the network to capture what we call "semantic profiles" of nodes, and then applies a distance-based outlier detection method to find abnormal nodes that are significantly different from their closest neighbors. To facilitate validation, we designed a novel explanation mechanism that can generate meaningful and human-understandable explanations for abnormal nodes discovered by our system. Such explanations not only facilitate the verification and screening out of false positives, but also provide directions for further investigation in determining the abnormal instances. The explanation system uses a classification-based approach to summarize the characteristic features of a node together with a path to sentence generator to describe these features in natural language.**

***Index Terms*– Data Mining, Machine Learning, Intrusion, Anomaly and Abnormal Instance**

I. INTRODUCTION

THE explosive increase in the number of networked machines and the widespread use of the internet in organizations have led to an increase in the number of unauthorized activities, not only by external attackers but also by internal sources, such as fraudulent employees or people abusing their privileges for personal gain or revenge. As a result, intrusion detection systems (IDSs) as originally introduced by Anderson [1] and later formalized by Denning [2], have received increasing attention in recent years. By definition intrusion detection is the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource. When Intrusion detection takes a preventive measure without direct human intervention, then it becomes an intrusion prevention system. A system that performs automated intrusion detection is called an Intrusion Detection System (IDS). Another important distinction is between systems that identify patterns of traffic or application data presumed to be malicious (misuse detection systems), and systems that compare activities against a 'normal' baseline (anomaly detection systems).Anomaly detection system (ADS) monitors the behavior of a system and flag significant deviations from the normal activity as an anomaly. Anomaly detection is used for identifying attacks in a computer networks, malicious activities in a computer systems, misuses in a Web-based systems. A network anomaly by malicious or unauthorized users can cause severe disruption to networks.

Therefore the development of a robust and reliable network anomaly detection system (ADS) is increasingly important. Traditionally, signature based automatic detection methods are widely used in intrusion detection systems. When an attack is discovered, the associated traffic pattern is recorded and coded as a signature by human experts, and then used to detect malicious traffic. However, signature based methods suffer from their inability to detect new types of attack. Furthermore the database of the signatures is growing, as new types of attack are being detected, which may affect the efficiency of the detection.

Data mining is a convenient way of extracting patterns, which represents mining implicitly stored in large data sets and focuses on issues relating to their feasibility, usefulness, effectiveness and scalability. It can be viewed as an essential step in the process of knowledge data discovery. Data are normally preprocessed through data cleaning, data integration, data selection, and data transformation and prepared for the mining task. Data mining can be performed on various types of databases and information repositories, but the kind of patterns to be found are specified by various data mining functionalities like class regression, association, classification, prediction, cluster analysis etc. Although data mining is a technology, companies are using powerful computers to shift through volumes of supermarket scanner data and analyze market research reports for years.

However, continuous innovations in computer processing power, disk storage and statistical software are dramatically increasing the accuracy of analysis, while driving down the cost we need to depend on data mining tools.

II. INTRODUCTION TO INTRUSION DETECTION

A) What is Intrusion Detection?

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network. Intrusions are caused by attackers accessing the systems from the Internet, authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and authorized users who misuse the privileges given them. Intrusion Detection Systems (IDSs) are software or hardware products that automate this monitoring and analysis process.

B) What is the Use of Intrusion Detection Systems?

Intrusion detection allows organizations to protect their systems from the threats that come with increasing network connectivity and reliance on information systems. Given the level and nature of modern network security threats, the question for security professionals should not be whether to use intrusion detection, but which intrusion detection features and capabilities to use. IDSs have gained acceptance as a necessary addition to every organization's security infrastructure. Despite the documented contributions intrusion detection technologies make to system security, in many organizations one must still justify the acquisition of IDSs.

There are several compelling reasons to acquire and use IDSs:

1) To prevent problem behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system,
2) To detect attacks and other security violations that are not prevented by other security measures,
3) To detect and deal with the preambles to attacks (commonly experienced as network probes and other "doorknob rattling" activities),
4) To document the existing threat to an organization
5) To act as quality control for security design and administration, especially of large and complex enterprises,
6) To provide useful information about intrusions that do take place, allowing improved diagnosis, recovery, and correction of causative factors.

C) Major Types of IDSs

There are several types of IDSs available today, characterized by different monitoring and analysis approaches. Each approach has distinct advantages and disadvantages. Furthermore, all approaches can be described in terms of a generic process model for IDSs.

1) Process Model for Intrusion Detection

Many IDSs can be described in terms of three fundamental functional components:

Information Sources: the different sources of event information used to determine whether an intrusion has taken place. These sources can be drawn from different levels of the system, with network, host, and application monitoring most common.

Analysis: the part of intrusion detection systems that actually organizes and makes sense of the events derived from the information sources, deciding when those events indicate that intrusions are occurring or have already taken place. The most common analysis approaches are *misuse detection* and *anomaly detection*.

Response: the set of actions that the system takes once it detects intrusions. These are typically grouped into active and passive measures, with active measures involving some automated intervention on the part of the system, and passive measures involving reporting IDS findings to humans, who are then expected to take action based on those reports.

2) Architecture

The architecture of IDS refers to how the functional components of the IDS are arranged with respect to each other. The primary architectural components are the Host, the system on which the IDS software runs, and the Target, the system that the IDS is monitoring for problems.

Host-Target Co-location: In early days of IDSs, most IDSs ran on the systems they protected. This was due to the fact that most systems were mainframe systems, and the cost of computers made a separate IDS system a costly extravagance. This presented a problem from a security point of view, as any attacker that successfully attacked the target system could simply disable the IDS as an integral portion of the attack.

Host-Target Separation: With the advent of workstations and personal computers, most IDS architects moved towards running the IDS control and analysis systems on a separate system, hence separating the IDS host and target systems. This improved the security of the IDS as this made it much easier to hide the existence of the IDS from attackers.

3) Goals

Although there are many goals associated with security mechanisms in general, there are two overarching goals usually stated for intrusion detection systems.

Accountability: Accountability is the capability to link a given activity or event back to the party responsible for initiating it. This is essential in cases where one wishes to bring criminal charges against an attacker. The goal statement associated with accountability is: *"I can deal with security attacks that occur on my systems as long as I know who did it (and where to find them.)"* Accountability is difficult in TCP/IP networks, where the protocols allow attackers to forge the identity of source addresses or other source identifiers. It is also extremely difficult to enforce accountability in any system that employs weak identification and authentication mechanisms.

Response: Response is the capability to recognize a given activity or event as an attack and then taking action to block or otherwise affect its ultimate goal. The goal statement associated with response is *"I don't care who attacks my system as long as I can recognize that the attack is taking place and block it."* Note that the requirements of detection are quite different for response than for accountability.

III. DATA MINING IDS

As an IDS can only detect known attacks, cannot detect insider attacks (privilege attacks), do not have holistic picture of the network to detect multi-step attacks over a long time period and though data for detection is available system administrations are limited, the better solution for an IDS can be Data Mining which is The process of extracting useful and previously unnoticed models or patterns from large data stores also called as "sense making".

To be more specific here are a few specific things that data mining might contribute to intrusion detection:

- Remove normal activity from alarm data to allow analysts to focus on real attacks
- Identify false alarm generators and "bad" sensor signatures
- Find anomalous activity that uncovers a real attack
- Identify long, ongoing patterns (different IP address, same activity)

To accomplish these tasks, data miners employ one or more of the following techniques:

- Data summarization with statistics, including finding outliers
- Visualization: presenting a graphical summary of the data
- Clustering of the data into natural categories
- Association rule discovery: defining normal activity and enabling the discovery of anomalies
- Classification: predicting the category to which a particular record belongs

A) Classification Techniques (Supervised Learning)

In a classification task in machine learning, the task is to take each instance of a dataset and assign it to a particular class. A classification based IDS attempts to classify all traffic as either normal or malicious. The challenge in this is to minimize the number of false positives (classification of normal traffic as malicious) and false negatives (classification of malicious traffic as normal). Five general categories of techniques have been tried to perform classification for intrusion detection purposes:

1) Inductive Rule Generation

The RIPPER System is probably the most popular representative of this classification mechanism. RIPPER [7], is a rule learning program. RIPPER is fast and is known to generate concise rule sets. It is very stable and has shown to be consistently one of the best algorithms in past experiments [8]. The system is a set of association rules and frequent patterns than can be applied to the network traffic to classify it properly. One of the attractive features of this approach is that the generated rule set is easy to understand; hence a security analyst can verify it. Another attractive property of this process is that multiple rule sets may be generated and used with a meta-classifier (Lee et al) [3], [5], [9], [4], [10]. Lee et al used the RIPPER system and proposed a framework that employs data mining techniques for intrusion detection.

This framework consists of classification, association rules, and frequency episodes algorithms that can be used to (automatically) construct detection models. They suggested that the association rules and frequent episodes algorithms can be effectively used to compute the consistent patterns from audit data. Helmer et al [6] duplicated Lee et al approach and enhanced it by proposing the feature vector representation and verifying its correctness with additional experiments.

Warrender et al [26] also used RIPPER to produce inductive rules and addressed issues that may arise if the mechanism was to be applied to an on-line system.

2) Neural Networks

The application of neural networks for IDSs has been investigated by a number of researchers. Neural networks provide a solution to the problem of modeling the users' behavior in anomaly detection because they do not require any explicit user model. Neural networks for intrusion detection were first introduced as an alternative to statistical techniques in the IDES intrusion detection expert system to model. In particular, the typical sequence of commands executed by each user is learned. Numerous projects have used neural nets for intrusion detection using data from individual hosts, such as BSM data [2]. McHugh et al have pointed out that advanced research issues on IDSs should involve the use of pattern recognition and learning by example approaches for the following two main reasons:

- The capability of learning by example allows the system to detect new types of intrusion.
- With earning by example approaches, attack "signatures" can be extracted automatically from labeled traffic data.

This basically eliminates the subjectivity and other problems introduced by the presence of the human factor. A different approach to anomaly detection based on neural networks is proposed by Lee et al. While previous works have addressed the anomaly detection problem by analyzing the audit records produced by the operating system, in this approach, anomalies are detected by looking at the usage of network protocols.

B) Clustering Techniques (Unsupervised Learning)

Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Machine learning typically regards data clustering as a form of unsupervised learning. Clustering is useful in intrusion detection as malicious activity should cluster together, separating itself from non-malicious activity. Clustering provides some significant advantages over the classification techniques already discussed, in that it does not require the use of a labeled data set for training.

Frank [12] breaks clustering techniques into five areas: hierarchical, statistical, exemplar, distance, and conceptual clustering, each of which has different ways of determining cluster membership and representation. Portnoy et al [13] present a method for detecting intrusions based on feature vectors collected from the network, without being given any information about classifications of these vectors. They designed a system that implemented this method, and it was able to detect a large number of intrusions while keeping the false positive rate reasonably low. There are two primary advantages of this system over signature based classifiers or learning algorithms that require labeled data in their training sets. The first is that no manual classification of training data needs to be done. The second is that we do not have to be aware of new types of intrusions in order for the system to be able to detect them. All that is required is that the data conform to several assumptions. The system tries to automatically determine which data instances fall into the normal class and which ones are intrusions.

Even though the detection rate of the system they implemented is not as high as of those using algorithms relying on labeled data, they claim it is still very useful. Since no prior classification is required on the training data, and no knowledge is needed about new attacks, the process of training and creating new cluster sets can be automated. In practice, this would mean periodically collecting raw data from the network, extracting feature values from it, and training on the resulting set of feature vectors. This will help detect new and yet unknown attacks. Eskin et al and Chan et al [14] have applied fixed width and k-nearest neighbor clustering techniques to connection logs looking for outliers, which represent anomalies in the network traffic. Bloedorn et al [11] use a similar approach utilizing k-means clustering. Marin et al employed a hybrid approach that begins with the application of expert rules to reduce the dimensionality of the data, followed by an initial clustering of the data and subsequent refinement of the cluster locations using a competitive network called Learning Vector Quantization. Since Learning Vector Quantization is a nearest neighbor classifier, they classified a new record presented to the network that lies outside a specified distance as a masquerader.

Thus, this system does not require anomalous records to be included in the training set. The authors were able to achieve classification rates, in some cases near 80% with misclassification rates less than 20%. Staniford et al [15] use “simulated annealing” to cluster events (anomalous packets) together, such that connections from coordinated port scans should cluster together. By using simulated annealing they reduce the run time from polynomial to linear. Marchette [16] used clustering to project high dimensionality data into a lower dimensional space where it could be more easily modeled using a mixture model. Sequeira and Zaki [56] also note the difficulty in determining the number of clusters in advance, and created the ”Dynamic Clustering” method to cluster similar user activity together, creating the proper number of clusters as it proceeds. Intrusion data are usually scarce and difficult to collect. Yeung et al [17] propose to solve this problem using a novelty detection approach.

In particular, they propose to take a nonparametric density estimation approach based on Parzen-window estimators with Gaussian kernels to build an intrusion detection system using normal data only. To facilitate comparison, they have tested their system on the KDD Cup 1999 dataset. Their system compares favorably with the KDD Cup winner which is based on an ensemble of decision trees with bagged boosting, as their system uses no intrusion data at all and much less normal data for training. Leung and Leckie [18] propose a new approach in unsupervised anomaly detection in the application of network intrusion detection. This new algorithm, called “fpMAFIA”, is a density based and grid based high dimensional clustering algorithm for large data sets. It has the advantage that it can produce clusters of any arbitrary shapes and cover over 95% of the data set with appropriate values of parameters. The authors provided a detailed complexity analysis and showed that it scales linearly with the number of records in the data set. They evaluated the accuracy of the new approach and showed that it achieves a reasonable detection rate while maintaining a low positive rate.

IV. MISUSE OR ANOMALY DETECTION ON DATA MINING TECHNIQUES

There are a variable number of intrusion detection systems where all of those intrusion detection systems can be placed into two major categories like misuse detection systems and anomaly detection systems. Machine learning algorithms are categorized using anomaly detection clustering and anomaly detection classification

A) Misuse Detection Using Supervised Learning

Misuse detection, based on binary or multiclass supervised learning methods, is an attractive candidate for IDS. A misuse detector generalizes from examples of known normal and hostile activity to derive a classifier for the two. The argument in favor of this approach is that many novel attacks are, in fact, minor variants on existing attacks, largely formulated to evade static signatures.

B) Anomaly Detection Using Supervised Learning

Under this view, a more promising approach is anomaly detection, in which only pure, “normal” data is used to train the system. Anomalies (a subset of which is attacks) are detected as significant deviations from this model of normal behavior. The arguments for this approach are that normal data is far easier to come by than are labeled attacks that a pure anomaly detector is unbiased toward any set of pre-trained attacks, and, therefore, that it may be capable of detecting completely novel attacks. The counterarguments are that hostile activities which appear similar to normal behavior are likely to go undetected, that it fails to exploit prior knowledge about a great many known attacks, and that, to date, false alarm rates for pure anomaly detection systems remain unusable high.

C) Misuse Detection Using Unsupervised Learning

As is known unsupervised learning is based not on the predefined training data set misuse detection is done mostly by using supervised learning and the unsupervised learning is not been preferred for misuse detection.

D) Anomaly Detection Using Unsupervised Learning

Applying unsupervised anomaly detection (also known as anomaly detection over noisy data [7]) Applying unsupervised anomaly detection in network intrusion detection is a new research area that has already drawn interest in the academic community. Eskin et al in 2002, investigated the effectiveness of three algorithms in intrusion detection:

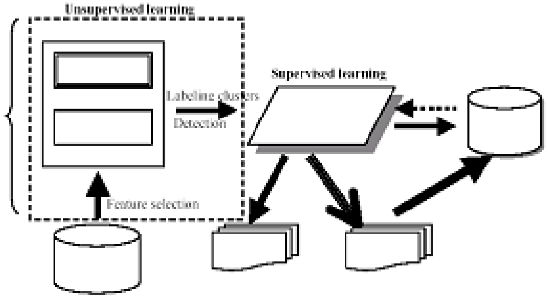

Fig. 1. Anomaly detection with machine learning

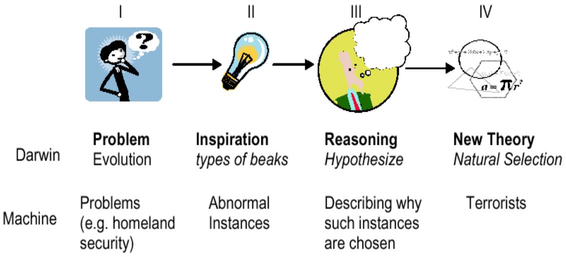

Fig. 2. Example of inspiration problem

The fixed-width clustering algorithm, an optimized version of the k-nearest neighbour algorithm, and the one class support vector machine algorithm. Old-meadow et al carried out further research based on the clustering method in (Eskin et al., 2002) and showed improvements in accuracy when the clusters are adaptive to changing traffic patterns. A different approach using a quarter sphere support vector machine is proposed in (Laskov et al., 2004), with moderate success. In (Eskin 2000), a mixture model for explaining the pres ence of anomalies is presented, and machine learning techniques are used to estimate the probability distributions of the mixture to detect anomalies. In (Zanero & Savaresi 2004), a novel two-tier IDS is proposed. The first tier uses unsupervised clustering to classify the packets and compresses the information within the payload, and the second tier used an anomaly detection algorithm and the information from the first tier for intrusion detection. Lane and Brodley in 1997, evaluated unlabelled data by looking at user profiles and comparing the activity during an intrusion to the activity during normal use.

E) Comparison of Unsupervised Over Supervised Learning in Anomaly Detection

Most current network intrusion detection systems employ signature-based methods or data mining-based methods which rely on labeled training data.(supervised). This training data is typically expensive to produce. Moreover, these methods have difficulty in detecting new types of attack. Using unsupervised anomaly detection techniques, however, the system can be trained with unlabelled data and is capable of detecting previously "unseen" attacks**.**

Algorithms have the major advantage of being able to process unlabeled data and detect intrusions that otherwise could not be detected. The goal of data clustering, or unsupervised learning, is to discovery a “natural” grouping in a set of patterns, points, or objects, without knowledge of any class labels.

We need a technique for detecting intrusions when our training data is unlabeled, as well as for detecting new and unknown types of intrusions.

V. PROBLEM STATEMENT

The central question we will pursue throughout this thesis is whether and how an AI program can model such a process to perform or assist humans to perform automatic discovery. To be more concrete, we focus on modeling the second and part of the third stage in the above discovery process as is shown in the figure. We develop a general framework that can automatically identify abnormal instances in data and explain them, with the goal to point out to humans a set of potentially interesting things in large, complex datasets. Note that there are three key features in this problem. First, our discovery targets data in the form of multi-relational networks or semantic graphs which allow the representation of complex relationships between objects of different types. Second, we are interested in abnormal nodes in these networks instead of central or important ones. Third, we want the discovery system to be able to explain its findings in a human-understandable form.

A) The Importance of Abnormal Instances

There are a variety of things one can discover from a network. For example, one can try to identify central nodes, recognize frequent subgraphs, or learn interesting network property. Centrality theory (Wasserman et al., 1994), frequent subgraph mining (Ramon et al., 2003) and small world phenomenon (Kleinberg, 2000) are among the well-known algorithms aimed at solving these problems. The goal of this work is different. We do not focus on finding central instances or pattern-level discovery. Instead we try to discover certain individuals or instances in the network that look different from others.

There are three reasons to focus on discovering these types of instances in an MRN. First, we believe that these kinds of instances can potentially play the “light bulb” role depicted in Figure 2, in the sense that something that looks different from others or from its peers has a higher chance to attract people’s attention or suggest new hypotheses, and the explanation of them can potentially trigger new theories. The second reason is that there are a number of important applications for a system that can discover abnormal nodes in an MRN. Finally, this is a very challenging problem and so far we are not aware of any system that can utilize the relational information in an MRN to perform anomaly detection.

B) UNICORN Algorithm

Unsupervised abnormal node discovery framework based on the methodologies and technique, we can now describe how our node discovery framework UNICORN identifies abnormal nodes in a multi-relational network.

*UNICORN(_EX*P) for the explanation system.

```
function UNICORN_EXP (Sp[][], n, 2_class, zero_sep, exp_num)
    array Lab[] := assign_label (Sp, n, 2_class);
    if (2_class)
        rules:=feature_sel (Sp, Lab, n, zero_sep,“outliers”,”normal”, exp_num);
    else{
        rules:= feature_sel(Sp_sub, Lab, n, zero_sep, “outliers”, “reference”, exp_num);
        forall k s.t. (Lab[k]=”outlier” or Lab[k]=”reference”)
            Lab[k]:=”outlier_reference”;
        rules:+= feature_sel (Sp_sub, Lab, n, zero_sep,”outlier_reference”, “normal”,
        exp_num );
    return NLG(rules)
Function assign_label(Sp, n, 2_class)
    array Lab[];
    if (2_clas )
        for k:= 0 to Sp.size-1
            if (near(Sp[k],Sp[n]))
                Lab[k]:=”outlier”;
            else
                Lab[k]:=”normal”;
    else
        g=get_gap_point(Sp,n);
        for k:= 0 to Sp.size-1
            if (near(Sp[k], Sp[g]))
                Lab[k]:=”reference”;
            else if (distance(Sp[k],Sp[g])<distance(Sp[n],Sp[g]))
                Lab[k]:=”outlier”;
            else Lab[k]:=”normal”
    return Lab;
function get_gap_point (Sp,n)
    array L[]= sort_distance (Sp,n);
    for m:= 1 to k //k is the maximum number of neighbors allowed for an outlier
        array gap[m-1]:=distance(Sp[n],Sp[L[m]])-distance(Sp[n],Sp[L[m-1]]);
    return argmax gap(x) // it returns the number x that maximizes gap(x)
function feature_sel (Sp, Lab, n, zero_sep, s1 , s2, exp_num)
    if (zer0_sep)
        Sp:=to_binary(Sp); //modify the feature values into binary values
    path DT:= decision_tree (Sp, Lab, s1, s2)
    return sub_path(DT, exp_num)
```

V. CONCLUSION

In this paper a general supervised and unsupervised for identifying abnormal instance in large and complex network datasets and an explanation mechanism to explain the normal or anomalies results was described. The specific approaches of the anomaly detection systems learning are characterized, we developed through our system performed on a representative natural dataset in the bibliography domains, we can also show that the UNICORN framework is domain independent and can be applied not only to identify suspicious instances in bibliographic networks, but also to find and explain abnormal or interesting instances in any multi-relational network. This leads to potential applications in a variety of areas such as scientific discovery, data analysis and data cleaning. Due to the generality of the techniques we developed they also lend themselves to other applications such as a novel outlier detection mechanism called explanation-based outlier detection, general node explanation to describe pertinent characteristics of arbitrary nodes, and abnormal path discovery to detect abnormal paths between nodes.

REFERENCES

[1]. Christos Douligeris, Aikaterini Mitrokotsa, ”DDoS attacks and defense mechanisms: classification and state-of-the-art”, Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 44, Issue 5 , pp: 643 - 666, 2004.

[2]. Ghosh, A. K., A. Schwartzbard, and M. Schatz,”Learning program behavior profiles for intrusion detection”, In Proc. 1st USENIX, 9-12 April, 1999

[3]. Lee, W. and S. J. Stolfo, ”Data mining approaches for intrusion detection”, In Proc. of the 7th USENIX Security Symp., San Antonio, TX. USENIX, 1998.

[4]. W. Lee, S.J.Stolfo et al, ”A data mining and CIDF based approach for detecting novel and distributed intrusions”, Proc. of Third International Workshop on Recent Advancesin Intrusion Detection (RAID 2000), Toulouse, France.

[5]. Lee, W., S. J. Stolfo, and K. W. Mok, ”A data mining framework for building intrusion detection models,” In Proc. of the 1999 IEEE Symp. On Security and Privacy, Oakland, CA, pp. 120132. IEEE Computer Society Press, 9-12 May 1999

[6]. Helmer, G., J.Wong, V. Honavar, and L. Miller, ”Automated discovery of concise predictive rules for intrusion detection”, Technical Report 99-01, Iowa State Univ., Ames, IA, January, 1999.

[7]. Cohen, W. W., ”Fast effective rule induction”, In A. Prieditis and S. Russell (Eds.), Proc. of the 12th International Conference on Machine Learning, Tahoe City, CA, pp. 115123. Morgan Kaufmann, 9-12 July, 1995.

[8]. S. Stolfo, A. L. Prodromidis and P. K. Chan, ”JAM: Java Agents for Meta- Learning over Distributed Databases”, in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, AAAI Press, Menlo Park, 1997.

[9]. Lee, W., S. J. Stolfo, and K. W. Mok, ” Mining in a data-flow environment: Experience in network intrusion detection,” In S. Chaudhuri and D. Madigan (Eds.), Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99), San Diego, CA, pp. 114124. ACM, 12-15 August 1999.

[10]. Lee, W., S. J. Stolfo, and K. W. Mok, ”Adaptive intrusion detection: A data mining approach,” Artificial Intelligence Review 14 (6), 533567, 2000.

[11]. Eric Bloedorn et al, ”Data Mining for Network Intrusion Detection: How to Get Started,” Technical paper, 2001.

[12]. Frank, J., ”Artificial intelligence and intrusion detection: Current and future directions”, In Proc. of the 17th National Computer Security Conference, Baltimore, MD. National Institute of Standards and Technology (NIST), 1994.

[13]. Portnoy, L., E. Eskin, and S. J. Stolfo, ”Intrusion detection with unlabeled data using clustering”, In Proc. of ACM CSSWorkshop on Data Mining Applied to Security (DMSA-2001), Philadelphia. ACM, 5-8 November, 2001.

[14]. Chan, P. K., M. V. Mahoney, and M. H. Arshad,"Managing Cyber Threats: Issues, Approaches and Challenges", Chapter Learning Rules and Clusters for Anomaly Detection in Network Traffic. Kluwer, 2003.

[15]. Staniford, S., J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans", Journal of Computer Security 10 (1-2), 105-136, 2002.

[16]. Marchette D., "A Statistical method for profiling network traffic", In First USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA, pp.119-128, USENIX, 9-12 April, 1999.

[17]. Yeung, D.-Y. and C. Chow, "Parzen-window network intrusion detectors", In Proc. of the Sixteenth International Conference on Pattern Recognition, Volume 4, Quebec City, Canada, pp. 385388. IEEE Computer Society, 11-15 August, 2002.

[18]. Leung, K. and Leckie, C., "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters", In Proc. Twenty-Eighth Australasian Computer Science Conference (ACSC2005), Newcastle, Australia, 1-3 February 2005, pp. 333-342, 2005.

M. Sathya Narayana completed B. Tech and Master of Technology in Computer Science and Engineering. He is presently working as Asst. Professor and HOD of IT Department in Visvesvaraya College of Engineering and Technology, Hyderabad, India. He is having 4-years of teaching experience and member in CSTA. Undergone CIT program conducted by IIIT Hyderabad, India.

B. V. V. S. Prasad completed MCA and Master of Technology in Computer Science Engineering. He is presently working as an Asst. Prof. in Visvesvaraya College of Engineering and Technology, Hyderabad, India. He is having about 4-years of teaching experience and an associate member of CSI and life member of ISTE. Undergone CIT program conducted by IIIT Hyderabad.

A. Srividhya completed B. Tech in IT and pursuing M. Tech in SE. She is presently working as Asst. Prof in SV College of Engineering and Technology. She has 3-years of teaching experience in this college and a member of IEEE. Undergone training in Information Security conducted by CDAC.

K. Pandu Ranga Reddy completed B. Tech in CSE and pursuing M. Tech in SE in Nishitha College of Engineering and Technology. He has 4-years of teaching experience.