An Intrusion Detection System based on Support Vector Machine using Hierarchical Clustering and Genetic Algorithm

Minakshi Bisen* & Amit Dubey**

* M.Tech Scholar, Department of Computer Science & Engineering, Oriental College of Technology, Bhopal, Madhya Pradesh, INDIA.
  E-Mail: er.mini27bisen[at]gmail[dot]com
** Head of the Department of Computer Science & Engineering, Oriental College of Technology, Bhopal, Madhya Pradesh, INDIA.
  E-Mail: amitdubey[dot]oriental[dot]ac[dot]in

Abstract—This study proposed an SVM based IDS which combines GA, Hierarchical Clustering and SVM technique. GA is used to preprocess the KDD Cup (1999) data set before SVM training. The proposed system reduces the training time and also achieve better classification of various types of attacks. GA provide the important feature and hierarchical clustering algorithm is used to provide a high quality, abstracted and reduced dataset to SVM for training. This system tries to increase accuracy of probe and u2r attacks. This system is implemented in MATLAB.

Keywords—Hierarchical Clustering; Genetic Algorithm; KDD Cup 1999; Network Intrusion Detection System; Support Vector Machine.

Abbreviations—Balanced Iterative Reducing using Clustering Hierarchies (BIRCH); Genetic Algorithm (GA); Network Intrusion Detection System (NIDS); Support Vector Machine (SVM).

I. INTRODUCTION

As the use of Internet is growing day by day, its security has been a focus in the current research. Nowadays, much attention has been paid to Intrusion Detection System (IDS) which is closely linked to the safe use of network services. Network Intrusion Detection System (NIDS), as an important link in the network security infrastructures, aims to detect malicious activities, such as denial of service attacks, port scans, or even attempts to crack into computers by monitoring network traffic. A common problems of NIDS is that it specifically detect known service or network attack only, which is called misuse, by using pattern matching approaches. On the other hand, an anomaly detection system detects attacks by building profiles of normal behaviors first, and then identifies potential attacks when their behaviors are significantly deviated from the normal profiles.

Many researches have applied data mining techniques in the design of NIDS. One of the promising techniques is Support Vector Machine (SVM), which solid mathematical foundations [Khan et al., 7] have provided satisfying results. SVM separates data into multiple classes (at least two) by a hyperplane, and simultaneously minimizes the empirical classification error and maximizes the geometric margin. Thus, it is also known as maximum margin classifiers. Hierarchical clustering algorithm that is used to produce fewer significant instances from a very large dataset. With fewer significant instances, the Support Vector Machines (SVMs) can achieve shorter training time and better classification performance.

An intrusion is unauthorized access or use of computer system resources. Intrusion detection systems are software that detects, identifies and responds to unauthorized or abnormal activities on a target system. The major functions performed by intrusion detection systems are: (1) monitor and analyze user and system activities, (2) assess the integrity of critical system and data files, (3) recognize activity patterns reflecting known attacks, (4) respond automatically to detected activities, and (5) report the outcome of the detection process [Burbeck & Simmin, 3]. Intrusion detection system can broadly be classified into misuse detection and anomaly detection. In Misuse detection, to identify intrusion well known attacks pattern or vulnerable spots in the system are used. While in anomaly detection, such attempts which are deviated from the normal established pattern can be recognized as intrusions. In Misuse detection, low false positive rate is obtained and minor variation from known
attacks cannot be detected while Anomaly detection has high false positive rate as it can detect novel attacks. In an ideal intrusion detection system, high attack detection rate along with 0% false positive rate should be there. This low rate of false positives is only achieved at the expense of ignoring minor malicious activity detection. As both of the this shows the complementary nature, many systems attempt to combine both techniques where misuse detection techniques can be used as the first line of defense, while the anomaly detection techniques can be as a second.

Most intrusion detection systems are classified as either a network-based or a host-based approach to recognize and detect attacks. A network-based intrusion detection system performs traffic analysis on a local area network. A host-based intrusion detection system places its reference monitor in the kernel/user layer and watches for anomalies in the system call patterns. The advantages of using network-based intrusion detection systems are no processing impact on the monitored hosts, the ability to observe network-level events, and monitoring an entire segment at once. However, as the complexity and capacity of networks increase, the performance requirements for probes can become prohibitive [Chen et al., 21]. Host-based intrusion detection systems can analyze all activities on the host, including its own network activities. Unfortunately, this approach implies a performance impact on every monitored system [Verwoerd & Hunt, 20].

This study proposed an intrusion detection system based on SVM, hierarchical clustering and genetic algorithm. Genetic algorithm is used to eliminate unimportant features from the training set so that the obtained SVM model could classify the network traffic data more accurately. Hierarchical clustering algorithm stores fewer abstracted data points of KDD Cup 1999 data set than the whole data set. Thus the system could greatly reduce the training time and achieve better detection performance in the resultant SVM classifier. The rest of this paper is organized as follows. Section 2 provides hierarchical clustering, genetic algorithm and SVM. Section 3 describes the proposed system. Section 4 represent the experimental results. Finally section 5 remarks the conclusion.

II. RELATED WORKS

Fuzzy Rough C-Means (FRCM), utilized the advantage of fuzzy set theory and rough set theory for network intrusion detection [Chimphlee et al., 4]. Performance of a comprehensive set of pattern recognitions and machine learning algorithms was analysed. Their system outperformed the KDD Cup 1999 winner’s system, combined several classifiers, one designated for one type of attacks in the KDD Cup 1999 dataset [Patcha & Park, 12]. Another fuzzy approach proposed, combined the neuro-fuzzy network, fuzzy inference approach and genetic algorithms to design their NIDS, and was evaluated by the KDD Cup 1999 dataset [Chen et al., 21]. Some researchers proposed a security vulnerability evaluation and patch framework, which enables evaluation of computer program installed on host to detect known vulnerabilities. Intruders can bypass the preventive security tools; thus, a second level of defence is necessary, which is constituted by tools such as anti-virus software and Intrusion Detection System (IDS) [Helmer & Liepins, 10]. The number of features extracted from raw network data, which an IDS needs to examine, is usually large even for a small network [Chou et al., 5]. Next-generation Intrusion Detection Export System (NIDES) was one of the few intrusion detection systems, which could operate in real-time for continuous monitoring of user activity or could run in a batch-mode for the periodic analysis of the audit data [Anderson et al., 1]. A stochastic clustering method, SCAN, which used expectation maximization to calculate the attribute value of the missing data, and also reduced the amount of data by feature selection [Lee et al., 8]. Some researchers used Genetic Algorithm (GA) for feature selection and SVM for intrusion detection [Patcha & Park, 12]. Based on BIRCH an incremental clustering intrusion detection algorithm, ADWICE was proposed. ADWICE can dynamically adjust the detection model, and has better incremental learning function, but is also very sensitive to the noise data similar to BIRCH [Burbeck & Simmin, 3]. An incremental clustering method based on the density to improve ADWICE was proposed [Fei et al., 6]. The implementation of SVMs requires the specification of the trade-off constant C as well as the type of the kernel function K. The choice of these parameters depends on the training data and consequently the set of independent variables (attributes) that enters the analysis is also an issue. The proposed methodology provides a framework to specify these parameters in an integrated context, using GAs [Satsiou et al., 2]. An ideal intrusion detection system is one that has a high attack detection rate along with a 0% false positive rate. However, such a low rate of false positives is only achieved at the expense of ignoring minor malicious activity detection. This provides an attacker with a small window of opportunity to perform arbitrary behaviors, giving them insight regarding the type of the intrusion detection system in use [Toosi & Kahani, 19]. Many recent approaches to intrusion detection systems utilize data mining techniques [Lam et al., 9]. These approaches build detection models by applying data mining techniques to large data sets of an audit trail collected by a system [Helmer & Liepins, 10]. At present, data mining algorithm applied to intrusion detection mainly has four basic patterns: association, sequence, classification and clustering [Sulaimana & Muhsinb, 11]. Building IDS having a small number of false positives is an extremely difficult task. In this paper we present two orthogonal and complementary approaches to reduce the number of false positives in intrusion detection by using alert postprocessing. The basic idea is to use existing IDSs as an alert source and then apply either off-line (using data mining) or on-line (using machine learning) alert processing to reduce the number of false positives [Pietraszek & Tanner, 18].
III. DISCUSSION

3.1. Support Vector Machine

An SVM is a supervised learning method which performs classification by constructing an N-dimensional hyperplane that optimally separates the data into different categories. In the basic classification, SVM classifies the data into two categories. Given a training set of instances, labeled pairs \{(x, y)\}, where y is the label of instance x, SVM works by maximizing the margin to obtain the best performance in classification [Patcha & Park, 12]. Support Vector Machine is a popular learning technique due to its high accuracy and performance in solving both regression and classification tasks. Although, the training time in SVM is computationally expensive task as the whole time is used in solving a problem. Many researches are carried out in SVM to reduce the training time such as chunking the problem, decomposition approach using iterative method etc.

SVM is originated from structural risk minimization (SRM) principle, which shorten the generalization error, i.e., true error on unseen examples. SVM mainly concerned with classes and separate the data in a hyperplane defined by a number of support vectors. These support vectors are the subset of training data used to define the boundary between two classes. In case, SVM cannot separate the data into two classes, it projects the data into high-dimensional feature space by using kernel function. This high dimensional feature space create a hyperplane which allows linear separation. The kernel function is very important in SVM as it helps in finding the hyperplane and support vectors. There may be various kernel functions such as linear, polynomial or Gaussian.

![Support Vector Machine](image)

**Figure 1: Support Vector Machine**

3.2. Genetic Algorithm

The implementation of SVMs requires the specification of the kernel function K. The choice of this parameter depends on the training data and consequently the set of independent variables (attributes) that enters the analysis is also an issue. The proposed methodology provides a framework to specify this parameters in an integrated context using GA’s. The first step for the implementation of a GA involves the specification of an appropriate coding for each possible solution. In the context considered in this study each, solution is defined by the attributes used for model development and the parameter required to define the kernel function. GA is used to select the appropriate features from the large data set.

3.3. BIRCH Hierarchical Clustering Algorithm

The BIRCH hierarchical clustering algorithm applied in this system was originally proposed by [Horng et al., 16]. BIRCH is different from other clustering techniques such as CURE, ROCK, Chameleon because it stores fewer abstracted data points than the whole dataset. Each abstracted point represents the centroid of a cluster of data points. Compared to CURE, ROCK, and Chameleon, the BIRCH clustering algorithm can achieve high quality clustering with lower processing cost. The advantages of BIRCH are as follows:

1. Constructs a tree, called a Clustering Feature (CF) tree, by only one scan of dataset using an incremental clustering technique.
2. Able to handle noise effectively.
3. Memory-efficient because BIRCH only stores a few abstracted data points instead of the whole dataset.

3.3.1. Clustering Feature (CF)

The concept of a Clustering Feature (CF) tree is at the core of BIRCH’s incremental clustering algorithm. Nodes in the CF tree are composed of clustering features. A CF is a triplet, which summarizes the information of a cluster.

3.3.2. CF Tree

A CF tree is a height-balanced tree with two parameters, branching factor B and radius threshold T. Each non-leaf node in a CF tree contains the most B entries of the form (CFi, child i), where 1 <=i<= 6 B and child i is a pointer to its ith child node, and CFi is the CF of a cluster pointed by the child i. A CF tree is a compact representation of a dataset, each entry in a leaf node represents a cluster that absorbs many data points within its radius of T or less. A CF tree can be built dynamically as new data points are inserted. The insertion procedure is similar to that of a B+-tree to insert a new data to its correct position in the sorting algorithm.

In KDD99 data set redundancy is amazingly high. Obviously, such a high redundancy certainly influences the use of data. By deleting the repeated data, the size of data set is reduced from 494,021 to 145,586. Furthermore, in order to make the data set more efficient, hierarchical clustering using BIRCH is used to reduce the dataset. Hierarchical clustering is a popular clustering algorithm which aims to partition different data samples into certain clusters by evaluating the smallest distance between data and clusters.

In the proposed system, firstly on the data set Birch and GA is applied, so that it can find out preprocessed and optimal dataset. Now, on this reduced and optimal data set, SVM is applied which classifies the network traffic data.
KDD CUP 1999 dataset is an extension of DARPA 98 dataset with a set of additionally constructed features in it. However, it does not contain some basic information about the network connections (e.g., start time, IP addresses, ports, etc.). The dataset was mainly constructed for the purpose of applying data mining algorithms. Therefore, we employed this dataset as a test bench for our algorithms. The dataset contains around 4900000 simulated intrusion records. The simulated attacks fell in one of the following four categories: DOS, R2L, U2R, and PROBE. There are a total of 22 attack types and 41 attributes (34 continuous and seven categorical). It seems that the whole dataset is too large. However, generally, only 10% subset is used to evaluate the algorithm performance. The 10% subset contains all 22 attack types. It is composed of all the low frequency attack records and the 10% of normal records and high frequency attack records, such as surf, neptune, portsweep and satan. These four types of the attack records occupy 99.51% of the whole KDDCUP99 dataset and 98.45% of the 10% subset [Sulaimana & Muhsinb, 11].

In KDD Cup 1999 data set, there are 4,898,431 and 311,029 records in training and test data set. In this data set, attack records were classified into four categories viz DoS, U2R, R2L and Probe. In the training set 19.85% were normal traffic and rest were attack traffic while for the test data set it contains 19.48% as the normal traffic and rest were attack traffic. There are 41 quantitative and qualitative features in each record of KDD data set [Patcha & Park, 12]. The result of proposed system on the KDD data set are as follow-

| Types of Attack | Attack Instances |
|-----------------|------------------|
| Dos             | 377              |
| Normal          | 185              |
| Probe           | 185              |
| R2L             | 234              |
| U2R             | 19               |

From the result we have seen that fp of u2r and probe is less due to which accuracy of this attack is more and also in about 1000 instances about 18.5% and 1.9% are probe and u2r attacks respectively.

V. CONCLUSION

In this study, an SVM based intrusion detection system which combines genetic algorithm, hierarchical clustering algorithm and SVM technique. The genetic algorithm and BIRCH hierarchical clustering technique is used for data preprocessing. The Birch hierarchical clustering provide highly qualified, abstracted and reduced data set to the SVM training. The famous KDD CUPP 1999 data set was used to evaluate the proposed system. Compared with other intrusion detection, this system showed better performance in the detection of various attacks. The future work is to apply the SVM with other Data preprocessing techniques.

REFERENCES

[1] D. Anderson, T.F. Lunt, H. Javitz, A. Tamaru & A. Valdes (1995), “Detecting Unusual Program Behavior using the Statistical Component of the Next-generation Intrusion Detection Expert System (NIDES)”, Menlo Park, CA, USA: Computer Science Laboratory, SRI International, SRI-CSL-95-06.

[2] A. Satsiou, M. Doumpos & C. Zopounidis, “Genetic Algorithms for the Optimization of Support Vector Machines in Credit Risk Rating”.

[3] K. Burbeck & N.Y. Simmin (2007), “Adaptive Real-Time Anomaly Detection with Incremental Clustering”, Information Security Technical Report, Vol. 12, No. 1, Pp. 56–67.

[4] W. Chimphlee, A.H. Abdullah, M.N. Md Sap, S. Srinoy & S. Chimphlee (2006), “Anomaly-based Intrusion Detection using Fuzzy Rough Clustering”, Proceedings of the International Conference on Hybrid Information Technology (ICFIT’06).

[5] T.S. Chou, K.K. Yen & J. Luo (2008), “Network Intrusion Detection Design using Feature Selection of Soft Computing Paradigms”, International Journal of Computational Intelligence, Vol. 4, No. 3, Pp. 196–208.

[6] R. Fei, L. Hu & H. Liang (2008), “Using Density-based Incremental Clustering for Anomaly Detection”, Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Vol. 3, Pp. 986–989.

[7] L. Khan, M. Awad & B. Thuraisingham (2007), “A New Intrusion Detection System using Support Vector Machines and Hierarchical Clustering”, The International Journal on Very Large Data Bases, Vol. 16, No. 4, Pp. 507–521

[8] J.H. Lee, S.G. Sohn, B.H. Chang & T.M. Chung (2009), “PKGVUL: Security Vulnerability Evaluation and Patch Framework for Package-based Systems”, ETRI Journal, Vol. 31, No. 5, Pp. 554–564.
Minakshi Bisen was born in Balaghat in 1988. She received the BE degree (with distinction) in computer science and engineering from Sagar Institute of Research and Technology, RGPV, Bhopal in 2011. She is currently pursuing M.Tech in computer science and engineering from Oriental College of Technology, RGPV, Bhopal. She has attended the national conference held in various institutes and presented papers in different research areas. Her research interests include network intrusion detection system and data mining.

Amit Dubey was born in Piparya. He received, M.Tech degree in Computer Science and Engineering from RITS, Bhopal and B.E. degree from SIST, Bhopal. Presently, he is an HOD of department of Computer Science & Engineering. His area of interest is data mining.

[9] K.Y. Lam, L. Hui & S.L. Chung (1996), “A Data Reduction Method for Intrusion Detection”, *Systems Software*, Vol. 33, Pp. 101–108.

[10] G. Helmer & G. Liepins (1993), “Statistical Foundations of Audit Trail Analysis for the Detection of Computer Misuse”, *IEEE Transactions on Software Engineering*, Vol. 19, Pp. 866–901.

[11] N. Sulaimana & O.A. Muhsinb (2011), “A Novel Intrusion Detection System by using Intelligent Data Mining in Weka Environment”, *Procedia Computer Science*, Vol. 3, Pp. 1237–1242.

[12] A. Patcha & J.M. Park (2007), “Network Anomaly Detection with Incomplete Audit Data”, *Computer Networks*, Vol. 51, No. 13, Pp. 3935–3955.

[13] S. Horng, M. Su, Y. Chen, T. Kao, R. Chen, J. Lai & C.D. Perkasa (2011), “A Novel Intrusion Detection System based on Hierarchical Clustering and Support Vector Machines”, *Expert Systems with Applications*, Vol. 38, No. 1, Pp. 306–313.

[14] P. Soto (2001), “The New Economy Needs New Security Solutions”, http://www.xcf.berkeley.edu/~paolo/ids.html.

[15] T. Pietraszek & A. Tanner (2005), “Data Mining and Machine Learning towards Reducing False Positives in Intrusion Detection”, *Information Security Technical Report*, Vol. 10, No. 3, Pp. 169–183.

[16] A.N. Toosi & M. Kahani (2007), “A New Approach to Intrusion Detection based on an Evolutionary Soft Computing Model using Neuro-Fuzzy Classifiers”, *Computer Communications*, Vol. 30, Pp. 2201–2212.

[17] T. Verwoerd & R. Hunt (2002), “Intrusion Detection Techniques and Approaches”, *Computer Communications*, Vol. 25, Pp. 1356–1365.

[18] W.-H. Chen, S.-H. Hsu & H.-P. Shen (2005), “Application of SVM and ANN for Intrusion Detection”, *Computers & Operations Research*, Vol. 32, No. 10, Pp. 2617–2634.

[19] T. Zhang, R. Ramakrishnan & M. Livny (1996), “BIRCH: An Efficient Data Clustering Method for Very Large Databases”, *Proceedings of the ACM SIGMOD (SIGMOD'96)*, Pp. 103–114.