A Novel Approach to Database Intrusion Detection Using Support Vector Machine with Fuzzy Clustering

Manaswini Pattanayak[1], Edi Laxmi [2]

[1]Asst. Professor, Einstein Academy of Technology & Management, Bhubaneswar
[2]Professor, Einstein Academy of Technology & Management, Bhubaneswar

ABSTRACT: *Countering threats to the databases from malicious access of sensitive information is an important area of research and yet remains a great challenge. Our strategy is to model an intrusion detection system that is capable of detecting anomalous user access requests to databases. The key idea is to identify the useful input features so as to learn the profile of users and applications interacting with a database, which helps the classifier model to recognize anomalies in real time. Current techniques used in database security are not able to cope with the dynamic nature of the attacker. The use of soft computing approaches in database intrusion detection is an appealing concept due to its robustness and low solution cost and better rapport with reality. In the current research, a new classification method is proposed using soft computing techniques− support vector machine with fuzzy clustering. The performance of the proposed database intrusion detection system is evaluated by testing with large scale transactions generated from stochastic models. The experimental results illustrate that the proposed system provides more accurate intrusion detection with low false alarm rate as compared to support vector machines without the application of fuzzy clustering*

KEYWORDS – *Databases, intrusion detection, fuzzy clustering, support vector machine, learning*

I. INTRODUCTION

With the rapid development of internet and communication technology, much attention is given to the security of database systems. Some of the data contained in these databases are quite sensitive in nature. Hence, organizations need to control access to such data for both internal users as well as external attackers. Although many different approaches are developed, still several loopholes are there in each of the existing systems. Almost all the classical security mechanisms are based on the auditing mechanism of the operating system and applications which are insufficient to describe which attributes are modified and whether it is legal to modify these data items at that time or not. Most importantly, these systems are alone not sufficient in securing sensitive information against novel attacks like insider threats.

AN insider threat is an authorized access by internal users who have legitimate access to the database but misuse their privileges. This could cause severe loss to an organization if the attack went undetected. The insider attack detection approaches are generally anomaly based and may range from rule based detection to statistical anomaly detection. Anomaly detection needs to maintain the past records of users' behaviors and the statistics for normal usages, which is referred to as “profiles”.

Anomaly intrusion detection identifies deviations from the normal usage behavior patterns to determine the behavior as legitimate or malicious. The challenging task is to construct the normal usage pattern for a database. Modeling and decision making in intrusion detection process falls into the category of classification problem. In recent years, the classification is generally done by applying several machines learning algorithms. In the current study, we explore the feasibility of applying Support Vector Machines (SVMs) to detect intrusive activities done by a user on a database system.

Furthermore, it is found that despite the use of various prevention based security mechanisms by database systems, intrusion detection is required as an additional security layer for protecting databasesby classifying incoming sequence of actions as malicious or genuine. However, very limited research has been carried out in the field of intrusion detection in the database. The Hidden Markov Model has been proposed in [1] to model the behavior of a user. Lee et al. [2] have used time signatures in discovering database intrusions. Their approach is to tag the time signature to data items. A security alarm is raised when a transaction attempts to write a temporal data object that has already been updated within a certain period. Another method presented by Chung et al. [3] identifies data items frequently referenced together and saves this information for later comparison. A data mining based IDS proposed by Srivastava et al. considers the sensitivity of attributes while

mining the dependency rules [4]. Panigrahi et al. proposed two stage database intrusion detection systems which consider inter and intra transactional features of database transactions along with the sensitivity level of attributes within a table[5,6].

This paper addresses the issue of identifying important database transaction features which can be useful in building user's normal profile that helps in uniquely identifying each user's behavioral pattern. The deviation from the normal patterns gives can indication of intrusive behavior, which can be detected by our proposed database intrusion detection system(DIDS).

II. PROPOSED APPROACH

The proposed approach deploys Support Vector Machines (SVMs) with Fuzzy Clustering to detect intrusive activities in databases [7]. In this research, Support Vector Machines are trained to model the normal behavior of a database user. Deviation from the normal patterns are detected by the DIDS and declared as malicious. To meet the above objectives, a comprehensive algorithm has been proposed as discussed in section (II.I), having two principal modules with the following major functionalities:

Data Reduction through Fuzzy Clustering

Intrusion Detection by SVM Classifier

Proposed Algorithm

Intrusion detector based on fuzzy clustering and Support Vector Machines

Input: Dataset consisting of genuine and malicious database transactions

Output: Classification results

Steps:

(1) Partition the dataset into two groups- training data and testing data

(2) Cluster the trained dataset by using fuzzy clustering.

(3) Determine the training parameter of SVM.

(4) Train the SVM classifier using the clusters

(5) For a given record (test data) , test it with the trained SVM classifier model.

(6) Return the detection results.

After having a detailed survey of the literature, it doesn't reveal any public availability of real dataset. In this research work, we have therefore used the simulator proposed in [5] which generates the synthetic transactions of both genuine user and intruders by considering appropriate mathematical distributions. The simulator follows the transactional web benchmark (TPC-W)[8] schema.

Once the database transactions are obtained, the necessary transaction features are extracted which are employed by our DIDS to detect anomalous behavior. In this investigation, the profile for each transaction is described by the following seven important transactional features:<*location_id, attribute_id_seq, command_id, relation_seq , time_slot, attribute_count, time_gap*>.

- *attribute_id_seq*: Sequence of attributes in a transaction
- *command_id*: A unique id is given to the most frequent commands used in the database:Create (1), Select (2), Insert (3), Alter (4), Update (5) and Delete (6).

1, 2, 3, etc. Subheadings are numbered 1.1, 1.2, etc. If a subsection must be further divided, the numbers 1.1.1, 1.1.2, etc.

- *time_slot*: Time slot in which a transaction occurs. We have partitioned 24 hrs in a day into 48 time slots, each of thirty minutes duration.
- *relation_seq*: Sequence of relations accessed in a transaction.
- *attrib_count:* Describes the count of the differenttypes of attributes accessed in a transactionbased on their sensitivity. The sensitivity of attributes signifies the priority of their vulnerability to intruders. It is categorized into three levels: high, medium and low and operation on these attributes are categorized into read and write operations. Accordingly, the attribute count is categorized into six groups:

(a) HSWC(High Sensitive Write Count):number of high sensitive attributes modified in a transaction.
(b) *HSRC*(High Sensitive Read Count):number of high sensitive attributes read in a transaction.
(c) *MSWC* (Medium Sensitive Write Count): number of medium sensitive attributes modified in a transaction.
(d) *MSRC* (Medium Sensitive Read Count): number of medium sensitive attributesread in a transaction.
(e) LSWC(Low Sensitive Write Count):number of low sensitive attributes modified in a transaction.

(f) *LSRC* (Low Sensitive Read Count): number of low sensitive attributes read in atransaction.

- *time_gap:* Time gap between current transaction and previous transaction done by the same user.

The following example of a transaction is taken into consideration which consists of two queries *q1* then *q2* submitted to perform a specific task.

q1: *Select a, b from table T1 and T2 where T1.c = T2.c*

q2: *Update T4 set d = 10 where e = 1*

where *a, b* and *c* are the attributes of tables *T1* and *T2* and *d* and *e* are the attributes of table *T4*. Suppose *a* and *d* are high sensitive attributes; *b, c* and *e* are low sensitive attributes.According to the values assigned to the various transaction features, the transaction is represented as follows: *<0,11,2,12,5,010020,5>*for *q1* and <1,45,5,45,5,100001,6> for *q2*. The above representation of query q1 describes that the transaction deals with select operation (command_id = 2) being carried out at the work place during the time_slot 5 (from 2am-2.30am) where a, b are first attribute of table T1 (id = 1) and T2 (id = 2) respectively with a time gap of 5 hours from the previous transaction.

The following section describes about the principal functionalities of the proposed DIDS.

B. Data Reduction through Fuzzy Clustering

Once a user profile is built, we partition the dataset to training and testing dataset. For better modeling of the intrusion detection system, the huge dataset of transactions should be reduced. For this reason, a clustering method is applied in this research. The clustering process considers the data tuples as objects. They partition the objects into groups or clusters so that objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters. We have used fuzzy clustering for reducing the dataset since the use of hard clustering is limited in real life applications. The fuzzy clustering method assigns a fuzzy membership value to each attribute present inside the cluster and the individual data point may belong to more than one cluster.

C. Intrusion Detection by SVM Classifier

The clustered data are given as input to the SVM classifier and a trained classifier model is generated which model the normal behaviour of users. Support Vector Machine is a soft computing tool which can recover underlying dependencies between the given inputs and outputs by using training dataset. It solves various classification problems by changing parameters that control how they learn as they cycle through training data [7, 9]. Moreover, SVMs can plot the training vectors in high-dimensional feature space, labelling each vector by its class. The classifier provides a generic mechanism to fit the surface of the hyper plane to the data through the use of a kernel function.

Following are some reasons which make the SVMs suitable for intrusion detection process in databases:

- Low expected probability of generalization of errors
- Faster execution speed
- SVMs are relatively insensitive to the number of data points as they can potentially learn from a larger set of patterns [9].

III. EXPERIMENTAL RESULTS

The correctness and efficiency of a classifier can be determined through its True Positive Rate (TPR), False Positive Rate (FPR), accuracy and Receiver Operating Characteristics (ROC) curve. TPR indicates the percentage of intrusive tuples that are correctly labelled by the model and FPR refers to the genuine samples that are mislabeled as intrusive.The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. The classifier returning the lowest error rate is the best classifier and the parameters used in training are the best tuning parameters which are being used to train the SVM classifier model. After the model is generated, the test dataset is used to find out the accuracy, TPR and FPR of the proposed DIDS. The ROC curve is a visual tool which shows the trade-off between the TPR and FPR of a given model.

The following table shows the comparative results on the proposed DIDS by applying fuzzy clustering(Fuzzy_SVM) and without clustering in SVMs based on the three parameters − Accuracy, TPR and FPR. From the results shown in Table 1, it is clearly understood that Fuzzy_SVM is showing the best performance in terms of all the three parameters.

TABLE 1 : PERFORMANCE EVALUATION OF SVM WITH UZZY_SVM

Type of SVM	Accuracy (in %)	TPR (in %)	FPR (in %)
SVM	90.16	80.0	7.01
Fuzzy_SVM	96.26	98.36	3.13

Fig. 1 and Fig. 2 represent the Receiver Operating Characteristic (ROC) curves of FUZZY_SVM and SVM respectively. It is observed from the curves that when the error rate tends towards zero, TPR in ROC curves is tending towards one while maintaining very low FPR.However,the number of iterations at which the model converged to zero is quite more in case of SVM without clustering compared to FUZZY_SVM.

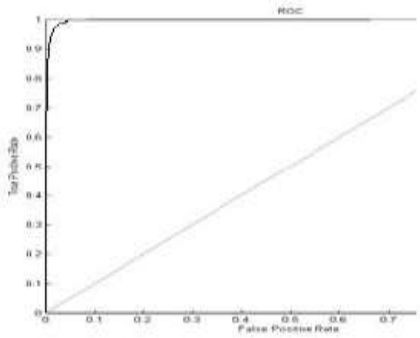

Fig 1. ROC Curve of FUZZY_SVM

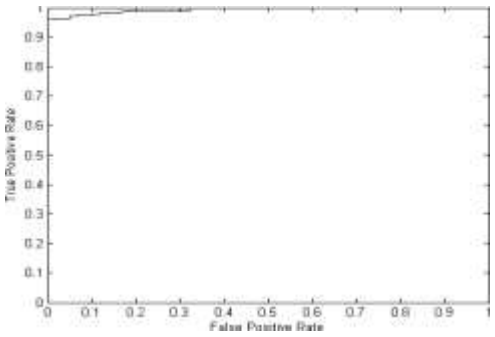

Fig. 2. ROC Curve of without Fuzzy Clustering

IV. CONCLUSIONS

Advancement in networking and electronic commerce has forced a change in the basic security design of database systems. Intrusion detection systems for the database has the potential advantages of reducing the manpower needed in monitoring, increasing detection efficiency and learn about new vulnerabilities. In this paper, we have introduced a novel approach for intrusion detection in databases by using SVM along with fuzzy clustering technique. Our results indicated that the use of SVM, along with fuzzy clustering performance is superior compared to that of SVM without clustering. In future research, other data mining techniques and soft computing tools may be applied for intrusion detection. Comparisons of various unsupervised and supervised learning mechanisms will provide clues for selecting appropriate models for effective database intrusion detection.

REFERENCES

[1] D. Barbara, R. Goel and S. Jajodia "Mining malicious data corruption with hidden markov models", Research Directions in Data and Applications security, IFIP , vol. 128,2003. pp. 175–189,2002

[2] V.C.S. Lee , J. Stankovic and S. Son, "Intrusion Detection in Real Time Databases via Time Signatures", Proceedings of the 6th IEEE Real-Time Technology and Applications Symposium (RTAS), pp. 124–133,2000

[3] C.Y. Chung , M. Gertz and K. Levitt , "DEMIDS: A Misuse Detection System for Database Systems", Proceedings of the Integrity and Internal Control in Information System, pp. 159–178,1999

[4] Srivastava, S. Sural, and A.K. Majumdar (2006). "Weighted intratransactional rule mining for database intrusion detection" LAdvances in knowledge discovery and Data mining, Lecture Notes in Computer Science, vol.3918,pp.611-620,2006

[5] S. Panigrahi, S. Sural, and A.K. Majumdar "Two-stage database intrusion detection by combining multiple evidence and belief update", Information Systems Frontiers, vol. 15 (Issue 1), pp. 35-53,2013

[6] S. Panigrahi, S. Sural, and A.K. Majumdar, " Detection of intrusive activity in databases by combining multiple evidences and belief update", In IEEE symposium on computationalintelligence in cyber security (CICS 2009) ,pp. 83–90,2009

[7] Jang J.S.R., (2004) "Neuro-fuzzy and soft computing", PHI publication, 1st edition, Pearson education, New Delhi.

[8] Transaction Processing Performance Council (2008). TPC Benchmark™ W (web commerce), specification, version 1.2.0. http://www.tpc.org/tpcw/default.asp.

[9] R.C. Garcia and J.A Copeland , (2000) "Soft computing tools to detect and characterize anomalous network behavior", IEEE Southeast conference, pp. 475-478.

