



Multi-dimensional and Multi-level Data Dependency with Attribute Sensitivity for Database Intrusion Detection

Indr Jeet Rajput*, Udai Pratap Rao
Department of Computer Engineering
SVNIT, Surat Gujarat, India

Abstract— here we propose database intrusion detection system and it works on data mining technique. In recent era the state of art in the field of database intrusion detection mechanisms where the researchers worked on single attribute, multiple attribute or sensitivity of attribute to generate the data dependency. Our proposed approach work is also based on data dependency rule generation from database transaction. But in our approach we consider both multi-level and multiple-dimensional data dependency with attribute sensitivity for identifying anomalous database transaction. Since data dependency reflect correlation between data item and if any transaction that does not follow these dependency rules are identified as malicious

Keywords— Data Mining, Database security, Intrusion detection system, support, confidence, Malicious transaction, Data dependency.

I. INTRODUCTION

Data represent today valuable asset for companies and organizations. With the developments of web application most of the work done on-line, the threat of security violations has also increased because number of entry point increase. The traditional method such as access control mechanism, authentication, encryption technologies and so forth are not much help when it comes to preventing data theft from intruders. In general there are two type of attacks inside and outside on the basis of the source form which it occurs [1]. Inside intruders are primarily the authorized users of an organization having a certain degree of access privileges on the system resource but he performs malicious actions. In contrast, outside refers to the damages caused by adversaries who are not valid users of organizations and are also not expected to be familiar with the system and security setup of the organizations. He attempts to first break in and then performs malicious actions. Deleting inside attack is usually more difficult compared to outside attack. For detecting this type of intrusion, a intrusion detection systems to be used. IDS are a second line of defence. Intrusion commonly defined as a set of actions that almost to be violet the integrity, confidentiality, and availability of a system. Intrusion detection is a process of identifying important events occurring in a system and analysing them for possible presence of intruder. IDS are software or hardware product that automates this monitoring and analysis process. IDS technique fall two categories [2]. Anomaly detection: it is attempt to identify malicious transaction based on deviation on the profile of a user's normal behaviour. It analysis user current behaviours and compared it with profile representing his normal behaviour. It is well suited for the detection of previously unknown attacks. It is have a high false positive rate. Misuse detection: it is the ability to identify intrusion based on known pattern for the malicious activity. A misuse detection models takes decision based on comparison of users session commands with rule or signature of attack previous usually used by attackers.

In this work we reveal a new scheme for identifying malicious transaction pattern to detect attack created either by insider or outsider intruder. For this purpose we use data mining technique such that data dependency rule generation approaches. Researcher has started using data mining technique in the emerging field of information and system security and specify in IDS. Data mining refers to extracted useful knowledge from huge volume of data. It is a composites technique from multiple disciplines such as database technology, statistics, machine learning, high-performance computing, spatial data mining analysis, neural network and others. In this approach we propose database IDS using multi-level and multi-dimensional data dependency with attribute sensitivity. Most of the work done by previous researcher consider data dependency on either single attribute, sensitive attribute, multi-level and multiple data dependency not consider both sensitivity and multi-level and multi-dimensional data dependency. The approach profiling multi-level and multi-dimensional data dependency with attribute sensitivity based on a training set of legitimate transactions in the database log. The transactions that do not follow the extracts data dependency rules are marked as malicious.

II. RELATED WORK

Various approaches have been proposed by researchers to tackle the problem of finding malicious database transactions using data mining technique [3]. The author proposed a data mining framework for construction of ID models. The key idea is to apply data mining program namely classification, Meta learning, association rules and frequent episodes to audit data for computing misuse and anomaly detection. Lee et al [4] author describes how the fingerprints for database transaction can be represented and presents an algorithm to learn and summarize SQL

statements into fingerprints. For database intrusion detection they summarize the incoming SQL query and compare with existing fingerprints, if it does not match, then it is represents as malicious. Bertino et al [5] proposed a database IDS that has similarity with role based access control (RBAC) models in profile granularity. They use C-tuple represented by query type, number of table accessed, number of attribute access, M-tuple query type, accessed table name, number of attribute accessed from individual table, and F-tuple by query type accessed table name, access attribute name to represent query. They build role profiles using classification, which is then used to detect anomalous behaviour. The approach Presented in [5] is query based which does not detect transaction level dependency resulting some of the database attacks may undetected. It can easily detect the attributes which are to be referred together, but it cannot detect the queries which are to be executed together. These problems resume by Rao et al. [17], this approach extracts the correlation among queries of the transaction. In this approach database log is read to extract the list of tables accessed by transaction and list of attributes read and written by transaction. They build role profiles using classification, which is then used to detect anomalous behaviour. Chung et al [6] is proposed an approach is that, the access pattern of users typically form some working scope with comprise sets of attribute. That is usually referred together with some value. DEMIDS use “working scope” to find frequent item sets. They define a relation of distance measures. That captures the closeness of set of attribute with respect to the working scope. It identifies data items frequently referenced together and identifies the working scope of users for detecting intrusions. Hu et al [7] use transaction level attribute dependency for detecting malicious transactions. The attribute depends with high support and confidence value form dependency rules. Transaction that does not follow any of the mined data dependency rules are marks as malicious. Data dependency refers to the access correlation among data items. That data dependency rules generated in the form of classification rules. Bandhakavi et al [8] author proposed approach is to detect anomalies SQL query structures. It dynamically mines the programmer-intended query structure on any input and detects attack by comparing it against the structure of the actual query issued. Lee et al [9] have proposed real-time database intrusion detection using time signatures. Real-time database has to deal with data those change its value with times. There temporal data objects are used to reflect the status of object. It tag the time signature to data items. A transaction attempt to write a temporal data object which has already been update within a certain period, an alarm is raised.

None of the related work explains above consider attribute sensitivity. They treat attribute of database with the similar sensitivity. The sensitivity of attribute in real world are not similar depend on its important in application. Hence therefore, they required to assign higher weightage to high sensitive attributes. Wang et al [10] have proposed a weighted association rule mining approach in which they assign numerical weight to each item to reflect intensity of the item within transaction. It finds the frequent item without considering weight and this apply weight for association rule generation. Tao et al [12] use weighted support for discovering the significant elements during the frequent element finding phase. In this paper address the issue of discovering significant binary relationship in transactional datasets in a weighted setting. Traditional model of association rule mining is adapted to handle weighted association rule mining problem when each item is allowed to have a weight. In order to tackle this challenge, we made adoption on the traditional association rules mining model under the “significant-weight support” metric framework instead of the “large-support” framework used in previous work. In this new proposed model the iterative generation and pruning of significant itemsets is justified by “weighted downward closer property”. A. srivastava et al [13] in this work we have identified some of the limitation of the existing intrusion detection system given, and then incapability in treating database attribute at different level of sensitivity in particular. In every database, some of the attribute are consider more sensitive to malicious modification compare to others. In this approach here with explain an algorithm. For finding dependency among important data items in a relational database management system, this approach generates more rules as compared to non-weighted approach. Srivastava et al [14] offered a weighted sequence mining approach for detecting database attacks. However, these models only consider sequential data dependencies and data dependency at a single granularity level i.e. attribute dependency.

None of the above approach consider multi-level and multi-dimensional data dependency model for identifying malicious database transactions. Hu et al [15] consider data dependency at different granularities level, such as attribute, relation, database or even distributed site. We can derive for detecting well crafted malicious transactions. Data dependency rules generate correlation among data items. We proposed multi-dimensional and multi-level data dependency with attribute sensitivity for data dependency. In this approach attribute sensitivity consider for both finding frequent itemsets and data dependency rule generation. In this paper we proposed multi-level and multi-dimensional dependency with sensitivity of attribute mining approach for profiling legitimate data access pattern from the database log directly. We have modified the work done by Hu et al [15].

III. MULTI-LEVEL AND MULTI-DIMENSIONAL DATA DEPENDENCY WITH ATTRIBUTE SENSITIVITY

A. Motivation

In present scenarios all the planning based on historical database. A huge amount of data to be stored in database, these data is accessed by number of users using web application. So that the number of entry point increase, which make database unsecured. Database consists of many attribute, it is very difficult for administrator to keep of attribute whether they are accessed or modify correctly or not. So we consider attribute sensitivity, depend on its important in database for intrusion detection. We also consider different granularity level, such as attribute, relation, or database for dependency rule generation. The proposed model is designed to identify malicious transaction submitted to the DBMS by an attacker that breaks the access control mechanism of a database system. Our model requires the database log records both read and write of each transaction. Here we proposed multi-dimensional and

multiple-level data dependency with attribute sensitivity for profiling legitimate data access patterns from the database directly.

B. Basic terminology

Sequence: A sequence is ordered list of attribute along with read and/or write operation perform on it. We denoted sequence as a Seq by $\langle O_1(a_1), O_2(a_2), \dots, O_n(a_n) \rangle$ where O_i an operation that can take values either 'r' or 'w' and a_1 to a_k is a attribute, $1 \leq i, k \leq n$.

Our approach considers data item sensitivity for finding malicious transaction. First we classified data item into three sensitivity groups depend on its important in database application for example, high sensitivity (HS), medium sensitivity (MS), and low sensitivity (LS). We need data dependency rules for these sensitive data item, if there is no rule for sensitive attribute, it is not possible to find malicious transactions. Since high sensitive data item access less frequently, so there is no dependency rule for that data item. So for increasing sensitivity of a data item we assign weight to each group.

Support: The support of a sequence is defined as the fraction of total transaction that contains the sequence. To perform frequent data item mining, we assign weight to each group based on their sensitivity. Weight of the group is the weight of the most sensitive data item present in a group. The weight assign to the group is used to calculate the support of each group in the transaction, are required in the second pruning step. Let there be a group G with weight W_G . Let N be the total number of transaction, if G is present n out of N transaction. Then the support of group G is:

$$\text{Support (G)} = (n * W_G) / N \tag{1}$$

Confidence: Let Dr be the dependency rule of the form $A \rightarrow B$, generated from group $G \in A, B$. Let Count(A), and Count(B) be the total count of the attribute A and that of B among the total number of transactions.

$$\text{Confidence} = \text{Support (AUB)} / \text{Support (A)} \tag{2}$$

where support are calculated by using formula (1).

C. Data Attribute and Set Dependency rules

Databases consist of many data attributes at different granularities. Our model work at different granularities, we define data attribute as a piece of information and Set as a collection of some data attributes.

Generation of Data Group: A data Group contain data attribute frequently accessed together and has atleast two data attribute. Every frequent element that contain only one data attribute has to be eliminated.

Data Attribute Dependency Rule Generation: For every data group one or more data dependency rules are generated in the form of

$$\{o_1(d_1.a_1), o_2(d_2.a_2), \dots, o_j(d_j.a_j)\} \rightarrow \{o_{j+1}(d_{j+1}.a_{j+1}), o_{j+2}(d_{j+2}.a_{j+2}), \dots, o_n(d_n.a_n)\} [s, c]$$

Where o, d, a, s, and c represent operation, group, data attribute, support and confidence respectively, only those rules are consider which have a confidence greater or equal to minimum confidence defined by user.

Consider the example transaction show in Table I [15]. In Table II, the three sensitivity groups and the weight of each attribute are shown. Given schema, we categories the data attribute into three sensitivity groups and assign numerical weight to each group. Let $w_1, w_2, w_3 \in R$, where R is set real number and $w_1 \geq w_2 \geq w_3$ are the weight of HS, MS, and LS respectively.

Table 1 Sample Transactions [15]

ID	Transactions
1	w(a.4), w(b.5), r(c.7), w(d.2)
2	r(b.5), w(d.3), w(c.7)
3	r(c.7), r(d.2), w(d.3), w(b.5), w(a.4), w(a.1), (d.6)
4	w(d.3), r(d.2), w(b.5), r(c.7), w(d.6), r(a.1)
5	r(c.7), w(d.3), w(b.5)
6	r(b.5), w(d.3), w(b.5), w(a.4), r(c.7), w(d.6)
7	R(d.3), r(d.2), r(c.7)
8	r(c.7), r(d.2), w(a.4), w(d.2)

Table II. Weight Table for the attribute used in the transaction

Sensitivity Group	Attribute	Weights
HS	1,3,7	3
MS	2, 5	2
LS	4, 6	1

We perform all the work into three steps, namely frequent data attribute generation, data group generation, and data attribute dependency rule generation.

Frequent Data Attribute Generation: The frequent data attribute generated by using apriori algorithm with data attribute sensitivity. We use Formula (1) for support calculation in Apriori algorithm [11]. We use a minimum support

threshold of 40% for finding frequent itemsets. Table III show frequent data attribute generated by using Apriori algorithm.

For finding data dependency, every sequence contains atleast two or more data attribute. So we eliminate every frequent itemset that contain only one data attribute. Table IV show generated data group.

Table III. Frequent Data Attribute

Frequent Itemsets	Weighted Support
r(d.2)	8
r(b.5)	4
r(c.7)	21
w(d.2)	4
w(d.3)	15
w(a.4)	4
w(b.5)	10
r(d.2), r(c.7)	12
r(d.2), w(d.3)	6
r(d.2), w(a.4)	4
r(d.2), w(b.5)	4
r(c.7), w(d.2)	6
r(c.7), w(d.3)	12
r(c.7), w(a.4)	12
r(c.7), w(b.5)	15
w(d.2), W(a.4)	4
w(d.3), w(a.4)	6
w(d.3), w(b.5)	12
w(a.4), w(b.5)	6
r(d.2), r(c.7), w(d.3)	6
r(d.2), r(c.7), w(a.4)	6
r(d.2), r(c.7), w(b.5)	6
r(d.2), w(d.3), w(b.5)	6
r(c.7), w(d.2), w(a..4)	6
r(c.7), w(d.3), w(a.4)	6
r(c.7), w(d.3), w(b.5)	9
r(c.7), w(a.4), w(b.5)	9
w(d.3), w(a.4), w(b.5)	6
r(c.7), w(d.3), w(a.4), w(b.5)	6

For each data group, one or more data attribute dependency rules are generated by using $A \rightarrow B$, if $\text{Support}(A \cup B) / \text{Support}(A)$ are greater than the minimum confidence. Table V show dependency rules generated by using minimum Confidence 80%. An algorithm used for generating dependency rules show in section 4. In this approach there are many redundant rule are generated. A rule R to be redundant if it has the same antecedent as another rule R^* and R consequent is a subset of R^* consequent. The problem with data attribute dependency rule generation is that it referred access together at very fine granularity. Because of this some of the data dependency missed out.

Table IV. Generated Data Group

Data Cliques	Wieghted Support
r(d.2), r(c.7)	12
r(d.2), w(d.3)	6
r(d.2), w(a.4)	4
r(d.2), w(b.5)	4
r(c.7), w(d.2)	6
r(c.7), w(d.3)	12
r(c.7), w(a.4)	12
r(c.7), w(b.5)	15
w(d.2), W(a.4)	4
w(d.3), w(a.4)	6
w(d.3), w(b.5)	12
w(a.4), w(b.5)	6
r(d.2), r(c.7), w(d.3)	6
r(d.2), r(c.7), w(a.4)	6
r(d.2), r(c.7), w(b.5)	6

r(d.2), w(d.3), w(b.5)	6
r(c.7), w(d.2), w(a.4)	6
r(c.7), w(d.3), w(a.4)	6
r(c.7), w(d.3), w(b.5)	9
r(c.7), w(a.4), w(b.5)	9
w(d.3), w(a.4), w(b.5)	6
r(c.7), w(d.3), w(a.4), w(b.5)	6

Table V. Data Attribute Dependency Rules

Data Item Dependency Rules	Weighted Confidence(%)
r(d.2)→r(c.7)	150
w(d.3)→r(c.7)	300
w(d.3)→w(b.5)	80
r(d.2), w(d.3)→r(c.7)	100
w(w.4)→r(d.2), r(c.7)	150
r(d.2), W(a.4)→r(c.7)	150
r(d.2), w(b.5)→r(c.7)	150
r(d.2), w(d.3)→w(b.5)	100
r(d.2), w(b.5)→w(d.3)	150
w(a.4)→r(c.7), w(d.2)	150
r(c.7), w(d.2)→w(a.4)	100
w(d.2)→w(a.4), r(c.7)	150
w(d.2), W(a.4)→r(c.7)	150
w(b.5)→r(c.7), w(d.3)*3	90
w(b.5)→ r(c.7), w(a.4)	90
w(d.3), W(a.4) →w(b.5)	100
r(c.7), w(d.3), w(a.4)→w(b.5)	100
w(d.3), w(a.4), w(b.5)→r(c.7)	100
w(a.4), w(b.5)→r(c.7), w(d.3)	100
w(d.3),w(a.4)→r(c.7), w(b.5)	100
w(a.4)→ r(c.7),w(d.3), w(b.5)	150

Group consists of many data attribute. For example group ‘a’ with data attribute 4 and write operation correlated with it and group ‘d’ with data attribute 6 and 3, write operation, with confidence 50% each. If a minimum confidence 80% then that rule not consider for finding malicious transactions. To solved the above problem by considering Set dependency rule generation.

Set Group: A Set Group referred to constructed by closely related Set that are access with same transaction. A form layout for Set Group is $\{o_1(d_1), o_2(d_2), \dots, o_n(d_n)\}$ where o_i represent read or write operation on Set d_i , $1 \leq i \leq n$. Weight of the set is the weight of attribute that present in set, which has a highest weight.

Set Dependency Rule Generation: The is represented in the form of

$$\{ o_1(d_1), o_2(d_2), \dots, o_j(d_j) \rightarrow o_{j+1}(d_{j+1}), o_{j+2}(d_{j+2}), \dots, o_n(d_n) \} [s, c]$$

Rule specify that data attribute in set in antecedent side are accessed in a transaction with operation $o_1 \dots o_j$ represent, the some data attribute in set consequent side are also access by an operation $o_{j+1} \dots o_n$ representing with same transaction with support s and confidence c . Here support is calculated by using formula (1). In this approach we consider only set, so that set with same operation on many time in transaction are replaced by one entity.

Using the same method which is used for data attribute dependency rules, we generated the Set dependency rules present in table IX with minimum confidence 80%. In this sequence of operation is irrelevant.

Table VI. Transformed Transactions

Train. ID	Transformed Transactions
1	w(a), w(b), r(c), w(d)
2	r(b), w(d), w(c)
3	r(c), r(d), w(d), w(b), w(a)
4	w(d), r(d), w(b), r(c), r(a)

5	r(c), w(d), w(b)
6	r(b), w(d), w(b), w(a), r(c)
7	r(d), r(c)
8	r(c), r(d), w(a), w(d)

Table VII. Frequent Itemsets from transformed transactions

Frequent Itemsets	Weighted Support
r(b)	4
r(c),	21
r(d)	8
w(a)	12
w(b)	10
w(d)	14
r(c), r(d)	12
r(b), w(d)	4
r(c), w(a)	12
r(c), w(b)	15
r(c), w(d)	18
r(d), w(a)	6
r(d), w(b)	4
r(d), w(d)	6
w(a), w(b)	9
w(a), w(d)	12
w(b), w(d)	10
r(c), r(d), w(a)	6
r(c), r(d), w(b)	6
r(c), r(d), w(d)	6
r(c), w(a), w(b)	9
r(c), w(a), w(d)	12
r(c), w(b), w(d)	15
r(d), w(a), w(d)	6
r(d), w(b), w(d)	4
w(a), w(b), w(d)	9
r(c), r(d), w(a), w(d)	6
r(c), r(d), w(b), w(d)	6
r(c), w(a), w(b), w(d)	9

Table VIII. Set Group

Set Group	Weighted Supports
r(c), r(d)	12
r(b), w(d)	4
r(c), w(a)	12
r(c), w(b)	15
r(c), w(d)	18
r(d), w(a)	6
r(d), w(b)	4
r(d), w(d)	6
w(a), w(b)	9
w(a), w(d)	12
w(b), w(d)	10
r(c), r(d), w(a)	6
r(c), r(d), w(b)	6
r(c), r(d), w(d)	6
r(c), w(a), w(b)	9
r(c), w(a), w(d)	12
r(c), w(b), w(d)	15
r(d), w(a), w(d)	6
r(d), w(b), w(d)	4
w(a), w(b), w(d)	9

r(c), r(d), w(a), w(d)	6
r(c), r(d), w(b), w(d)	6
r(c), w(a), w(b), w(d)	9

Table IX. Set Dependency rules

Set dependency Rules	Weighted Confidence(%)
r(d)→r(c)	150
r(c)→w(d)	85
w(d)→r(c)	85
r(b)→w(d)	100
w(a)→r(c), w(d)	100
w(a), w(d)→r(c)	100
r(c), w(a)→w(d)	100
r(c), w(b)→w(d)	100
w(b), w(d)→r(c)	100
r(c), w(d)→w(b)	83
r(c), r(d), w(d)→w(a)	100
r(c), r(d), w(a)→w(d)	100
r(d), W(a), w(d)→r(c)	100
r(d), w(a)→r(c), w(d)	100
r(c), r(d), w(d)→w(b)	100
r(d), w(b), w(d)→r(c)	100
r(d), w(b)→r(c), w(d)	100
r(c), r(d), w(b)→w(d)	100
w(b)→r(c), w(a), w(d)	90
r(c), W(a), w(b)→w(d)	100
w(a), w(b), w(d)→r(c)	100
w(a), w(b)→r(c), w(d)	100

D. Data Attribute and Set access sequence rules

The data attribute and Set dependency rule generated above not consider the sequence of operation in the rule. Data access by some legitimate transaction in a certain sequence. The sequential access pattern can be employed to increase the detection rate of malicious transaction.

Data Attribute Access Sequence: It is an arrange list of data attribute that are often access in sequence in a database transaction. It is represent in the form of $\langle o_1(d_1.a_1), o_2(d_2.a_2), \dots, o_n(d_n.a_n) \rangle$, where o_i is an operation, $d_i.a_i$ represent data attribute a_i in Set d_i $1 \leq i \leq n$.

Data Attribute Access Sequence Rule: A data access sequence rule define as either of them

$\langle o_1(d_1.a_1), o_2(d_2.a_2), \dots, o_j(d_j.a_j) \rangle \rightarrow \langle o_{j+1}(d_{j+1}.a_{j+1}), o_{j+2}(d_{j+2}.a_{j+2}), \dots, o_n(d_n.a_n) \rangle [s, c]$.

$\langle o_1(d_1.a_1), o_2(d_2.a_2), \dots, o_j(d_j.a_j) \rangle \leftarrow \langle o_{j+1}(d_{j+1}.a_{j+1}), o_{j+2}(d_{j+2}.a_{j+2}), \dots, o_n(d_n.a_n) \rangle [s, c]$.

Above specify two rule are not same except when support ($\langle A \rangle$) = support ($\langle B \rangle$).

Table X show frequent sequence mined from transactions in table I using sequence pattern mining algorithm such as AprioriAll[16] using formula (1) for support calculation with minimum support 40%.

Table X. Data Access Sequence frequent Attribute

Frequent Attribute	Weighted Support
r(d.2)	8
r(b.5)	4
r(c.7)	21
w(d.2)	4
w(d.3)	15
w(a.4)	4
w(b.5)	10
r(d.2), r(c.7)	6
r(d.2), w(a.4)	4
r(d.2), w(b.5)	4
r(b.5), w(d.3)	6
r(c.7), r(d.2)	6
r(c.7), w(d.2)	6
r(c.7), w(d.3)	6
r(c.7), w(a.4)	6
r(c.7), w(b.5)	6

w(d.3), r(c.7)	6
w(d.3), w(a.4)	6
w(d.3), w(b.5)	9
w(a.4), r(c.7)	6
w(a.4), w(d.2)	6
w(b.5), r(c.7)	9
w(b.5), w(a.4)	4
r(c.7), r(d.2), w(a.4)	6
r(c.7), w(d.3), w(b.5)	6
w(d.3), w(b.5), r(c.7)	6
w(d.3), w(b.5), w(a.4)	6

Table XI. Data Access Sequence Rules

Data access Sequence Rules	Weighted Confidence(%)
r(b.5)→w(d.3)	150
r(c.7)←w(d.2)	150
w(d.3)→w(b.5)	80
w(a.4)→w(d.2)	150
w(a.4)←w(d.2)	150
w(b.5)→r(c.7)	90
r(c.7)←r(d.2), w(a.4)	150
r(d.2)←r(c.7), w(a.4)	100
w(a.4)→r(c.7),r(d.2)	150
w(a.4)←r(c.7),r(d.2)	100
r(c.7), r(d.2)→w(a.4)	100
r(c.7), r(d.2)←w(a.4)	150
r(d.2), w(a.4)→r(c.7)	150
r(c.7), w(a.4)→r(d.2)	100
w(d.3)←r(c.7), w(b.5)	100
w(b.5)←r(c.7), w(d.3)	100
r(c.7), w(d.3)→w(b.5)	100
r(c.7), w(b.5)→w(d.3)	100
w(b.5)←w(d.3), r(c.7)	100
w(d.3),r(c.7)→w(b.5)	100
w(d.3)←w(b.5), w(a.4)	150
w(b.5)←w(d.3), w(a.4)	100
w(a.4)←w(d.3), w(b.5)	150
w(d.3),w(b.5)←w(a.4)	150
w(d.3), w(a.4)→w(b.5)	100
w(b.5), w(a.4)→w(d.3)	150

For data access sequence rules are generated by using algorithm section 4. The confidence is calculated by using formula (2), in which weight of attribute to be consider. Table XI contains data access sequence rules generated from the frequent sequence of table X

Set Access Sequence Rule: The Set access sequence rule specify on one of the two forms:

$\langle o_1(d_1), o_2(d_2), \dots, o_j(d_j) \rangle \rightarrow \langle o_{j+1}(d_{j+1}), o_{j+2}(d_{j+2}), \dots, o_n(d_n) \rangle [s, c]$

$\langle o_1(d_1), o_2(d_2), \dots, o_j(d_j) \rangle \leftarrow \langle o_{j+1}(d_{j+1}), o_{j+2}(d_{j+2}), \dots, o_n(d_n) \rangle [s, c]$

The method for generation of Set access sequence rule is an first convert every operation $o_i(d_i, a_i)$ in $o_i(d_i)$. It is differ from transformed transaction in Set dependency rule, in it duplicate can not be removed. This is because removing duplicate Set operation eliminates important sequential data access information. Second, frequent Set sequence is generated using a sequential pattern mining algorithm. Finally Set sequence rules are generated. In case of data Set duplicate entry in transaction removed where as duplicate entry kept on in table XII. Table XIII contains the frequent Set sequences generated by using algorithm AprioriAll [16] with minimum support 40%. We use formula (1) for calculating support of Set access sequences. Table XIV show the Set access sequence rule generated with minimum confidence 80%. For example $\langle w(b) \rightarrow w(d) \rangle$ specifies that if some data attribute in set 'b' is updated, some data attribute in set 'd' have to be updated afterward.

Table XII. Set Access Sequences

Tran. ID	Set Access Sequences
1	w(a), w(b), r(c), w(d)
2	R(b), w(d), w(c)

3	r(c), r(d), w(d), w(b), w(a), w(a), w(d)
4	w(d), r(d), w(b), r(c), w(d), r(a)
5	r(c), w(d), w(b)
6	r(b), w(d), w(b), w(d) w(a), r(c), w(d)
7	r(d), r(d), r(c)
8	r(c), r(d), w(a), w(d)

Table XIII. Frequent Set sequence

Frequent Set Sequence	Weighted Support
r(b)	4
r(c)	21
r(d)	12
w(a)	12
w(b)	10
w(d)	21
r(b), w(d)	6
r(c), r(d)	6
r(c), w(a)	6
r(c), w(b)	6
r(c), w(d)	18
r(d), r(c)	6
r(d), w(a)	6
r(d), w(b)	6
r(d), w(d)	9
w(a), r(c)	6
w(a), w(d)	12
w(b), r(c)	9
w(b), w(a)	6
w(b), w(d)	12
w(d), r(c)	6
w(d), w(a)	6
w(d), w(b)	12
r(c), r(d), w(a)	6
r(c), r(d), w(d)	6
r(c), w(a), w(d)	6
r(c), w(d), w(b)	6
r(d), w(a), w(d)	6
r(d), w(b), w(d)	6
w(a), r(c), w(d)	6
w(b), r(c), w(d)	9
w(b), w(a), w(d)	6
w(d), w(b), r(c)	6
w(d), w(b), w(a)	6
r(c), r(d), w(a), w(d)	6

Table XIV. Set Access Sequence Rules

Set Access Sequence Rules	Weighted Confidence(%)
r(b)→w(d)	100
r(c)→w(d)	85
w(a)→w(d)	100
r(c)←r(d), w(a)	100
w(d)←r(c), w(b)	100
r(c), w(b)→w(d)	100
w(d)←r(d), w(b)	100
r(d), w(b)→w(d)	100
w(d)←w(a), r(c)	100
w(a), r(c)→w(d)	100

w(b)→r(c), w(d)	90
w(d)←w(b), r(c)	100
w(b), r(c)→w(d)	100
r(c), w(d)←w(b)	90
w(d)←w(b), w(a)	100
w(b), w(a)→w(d)	100
w(b)←w(d), r(c)	100
w(d), r(c)→w(b)	100
w(d)←w(b), w(a)	100
w(b)←w(d), w(a)	100
w(d), w(a)→w(b)	100
w(b), w(a)→w(d)	100
r(c)←r(d), w(a), w(d)	100
r(d)←r(c), w(a), w(d)	100
w(a)←r(c), r(d), w(d)	100
w(d)←r(c), r(d), w(a)	100
r(c),r(d)→ w(a), w(d)	100
r(c), w(a)→r(d), w(d)	100
r(c), w(d)←r(d), w(a)	100
r(d), w(a)→r(c), w(d)	100
r(d), w(d)← r(c), w(a)	100
w(a), w(d)←r(c), r(d)	100
r(c), r(d), w(a)→w(d)	100
r(c), r(d), w(d)→w(a)	100
r(c), w(a), w(d)→r(d)	100
r(d), w(a), w(d)→r(c)	100

IV. ALGORITHMS

Algorithm 1: Data Attribute Dependency Rule Mining Algorithm

Input: A set of legitimate transactions from database log T, minimum support s, and minimum confidence c.

Output: Data Attribute dependency rule set DR.

Initialize data group DG, i.e., DG = { }

Initialize data attribute dependency rule set DR , i.e., DR = { }

Generate frequent attribute sets F using a frequent attribute set mining algorithm with support calculated by formula (1) from database log T

For each f ∈ F

If (|f| = 1)

F=F-f

DG=F

For each data group d ∈ DG

For each subset s of data group d

$$\text{if } \frac{\text{Support (Count (d) * Weight (d))}}{\text{Support (Count (s) * Weight (s))}} > c$$

DR=DR U {s→d-s}

For each rule r ∈ DR

If r has the same antecedent as another rule and r's consequent is the subset of that rule, i.e., r is a redundant rule

DR=DR -r

Algorithm II: Data access sequence Rule Mining Algorithm

Input: A set of legitimate transactions from database log T, data attribute dependency rules DR, minimum support s, and minimum confidence c.

Output: Data access sequence rule set SR, modified data attribute dependency rule set DR.

Initialize data access sequence rule set SR, i.e., SR = { }

Generate frequent sequences Q using a sequential pattern mining algorithm [16] with using formula (1) from database log T

For each frequent sequence q ∈ Q

If (|q| = 1)

Q=Q-q

For each sequence q ∈ Q

For each subsequence p ∈ q

For each subsequence | ∈ q, where every operation of | occurs before every operation in p

$$\text{if } \frac{\text{Support (Count (<l, p >) * Weight (<l, p >))}}{\text{Support (Count (p) * Weight (p))}} \geq C$$

SR=SR U {l←p}

For each subsequence r ∈ q, where every operation of r occurs after every operation in p

$$\text{if } \frac{\text{Support (Count (<p, r >) * Weight (<p, r >))}}{\text{Support (Count (p) * Weight (p))}} \geq C$$

SR= SR u {p → r}

For each rule x ∈ DR

If there exists a rule Y ∈ SR, where y's antecedent and consequent are the same as those of x respectively

DR=DR-x

For producing Set dependency rule and Set sequence rules, we first transferred database log from finer level of granularities from database show in table I, so only information left to be Set level, then apply Algorithm I and II for producing Set dependency and Set access sequence rules respectively.

V. COMPARISONS

We compare our proposed system with existing system implemented by Hu et al. [15] on the following metrics.

True Negative: Here total no. of genuine transaction 8 existing system detect only 6 out of 8 genuine transaction as normal but our system detect 7 out of 8 genuine transaction as normal so % of true negative is higher of our system as compared to existing system [15].

False Positive: Existing system detects 2 out of 8 genuine transactions as attack i.e. 25% false positive. But our system detects only 1 out of 8 genuine transactions as attack i.e. 12.5% false positive.

True Positive: It is explained with example let intrusion transaction r (c.7), r (d.2), w (d.3), w (b.5), w (d.2). In existing system [15] there is no rule for modification done by w (d.2) then it is allowed transaction as genuine. But our proposed system having the rule for modification done by w (d.2) and then it does not satisfy the rule therefore given transaction is a malicious transaction .

VI. CONCLUSIONS

In this paper we proposed data mining approach for detecting malicious transactions in database systems. Multi-dimension and multi-level data mining with attribute sensitivity are employed to find data attribute dependency rules, data access sequence rules, set dependency rules and set access sequence rules from the legitimate transactions. The proposed database intrusion detection system generates more rules as compared to non-weighted approach [15].

ACKNOWLEDGMENT

I will thank my teacher assistant professor Udai Pratap Rao and Dr. Dhiren R. Patel here for fervent help when I have some troubles in paper writing. I will also thank my class mates in laboratory for their concern and support both in study and life.

REFERENCES

- [1] Heady, R., Luger, G., Maccabe, A., Servilla, M.: "The Architecture of a Network Level Intrusion Detection System". Technical report, Computer Science Department, University of New Mexico (1990)
- [2] T. Bhavani et al., "Data Mining for Security Applications," Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing - Volume 02, IEEE Computer Society, 2008.
- [3] L. Zenghui, L. Yingxu, "A Data Mining Framework for Building Intrusion Detection Models Based on IPv6," Proceedings of the 3rd International Conference and Workshops on Advances in Information Security and Assurance. Seoul, Korea, Springer-Verlag, 2009.
- [4]. Lee, S.Y., Low,W.L.,Wong, P.Y.: "Learning Fingerprints for a Database Intrusion Detection System". In: Proceedings of the 7th European symposium on research in computer security, pp. 264–280 (2002)
- [5]. Bertino, E., Terzi, E., Kamra, A., Vakali, A.: "Intrusion Detection in RBAC-Administered Databases". In: Proceedings of the 21st annual computer security applications conference (ACSAC), pp. 170–182 (2005)
- [6]. Chung, C.Y., Gertz, M., Levitt, K.: "DEMIDS: A Misuse Detection System for Database Systems". In: Proceedings of the integrity and internal control in information system, pp. 159–178 (1999)
- [7]. Hu, Y., Panda, B.: "A Data Mining Approach for Database Intrusion Detection". In: Proceedings of the ACM symposium on applied computing, pp. 711–716 (2004)
- [8] Bandhakavi, S., Bisht, P., Madhusudan, P., and Venkatakrishnan V.: "CANDID: Preventing sql injection attacks using dynamic candidate evaluations". In the Proceedings of the 14th ACM Conference on Computer and Communications Security (2007)
- [9] Lee, V. C. S., Stankovic, I. A, and Son, S. H.: "Intrusion Detection in Real-time Database Systems Via Time Signatures". In the Proceedings of the 6th IEEE Real-Time Technology and Applications Symposium (2000)
- [10] W. Wang, J. Yang, P. S. Yu, "Efficient Mining of Weighted Association Rules", Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 270-274 (2000).
- [11] Agrawal, R., Imielinski, T., Swami, A. "Mining association rules between sets of items in large databases", In proceedings of the 1993 ACM SIGMOD international conference on management of data (1993).

- [12] F. Tao, F. Murtagh, M. Farid, "Weighted Association Rule Mining using Weighted Support and Significance Framework", Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 661-666 (2003).
- [13] A. Srivastava, S. Sural, A.K. Majumdar, "Weighted Intra-transactional Rule Mining for Database Intrusion Detection", Lecture Notes in Artificial Intelligence, Springer Verlag, Proceedings of Pacific-Asia Conference in Knowledge Discovery and Data Mining, pp. 611-620 (2006).
- [14] Srivastava, A, Sural S., and Majumdar, AK.: "Database Intrusion Detection Using Weighted Sequence Mining", Journal of Computers, vol.1, no. 4 (2006)
- [15] Hu, Y., Campan, A., Walden, J., Vorobyeva, I., Shelton, J.: "An effective log mining approach for database intrusion detection", In proceedings of systems man and cybernetics (SMC) 2010 IEEE international conference.
- [16] Agrawal, R. and Srikant, R.: "mining sequential pattern", In Proceedings of the 1995 international conference on data engineering, Taipei, Taiwan (1995).
- [17] Rao, U.P., sahani, G.J., Patel, D.R., "Detection of Malicious Activity in Role Based Access Control (RBAC) Enabled Databases". In Proceeding of Journal of Information Assurance and Security 5 (2010) 611-617.