International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database

Abdel Rahman Mahmoud[1], Dr. Nagy Ramadan[1], Abdel Moniem Helmy[1]
[1]Information Systems Department, Institute of Statistical Studies and Research (ISSR), Cairo University, Cairo, Egypt-12613
Email address: aymanmahmoud815@yahoo.com

Abstract-*The discovery of the association relationship among huge temporal database has been legendary to be helpful in selective selling, call analysis and business management. The current algorithms of association rule mining has limitations in handling temporal data in different data sets for these two main reasons. The first one lack of consideration of the exhibition period of each individual Item; the second reason is the lack of an equitable support counting basis for each item. The area of temporal data processing has much attention within the last decade as a result of the time-related features of the information; one will extract abundant important data that cannot be extracted by the general methods of data mining. However Most of the data mining techniques perform to treat temporal data as an unordered collection of events, ignoring its temporal information. The work proposed in this paper applies knowledge discovery techniques on a series of huge datasets obtained over a partition that contains a large number of transactions in the consecutive time period, instead of applying on the whole database. In addition, Instead of extracting rules throughout the whole timeline, we will extract the rules for consecutive time intervals with different time granularities. The result for that will be developing more efficient approach for mining temporal association rules on large data sets.*

Keywords-*Temporal Data Mining, Association Rule Mining, Database Knowledge Discovery.*

I. INTRODUCTION

Temporal Data Mining (TDM) is determined as the activity of looking for interesting correlations or patterns in large temporal datasets. Temporal data mining is an important extension of the data mining and it is a non-trivial extraction of implicit, potentially useful and previously unrecorded information with an implicit or explicit temporal content from a large database. Temporal data has importance in a category of fields, such as biomedicine, geographical data processing, financial data forecasting and Internet site usage monitoring. Temporal Data Mining is a rapidly evolving area of research that is at the intersection of several disciplines, including statistics, temporal pattern recognition, temporal databases, optimization and parallel computing [1-3].

Basically, temporal data mining is concerned with the analysis of temporal data and finding temporal patterns and regularities in sets of temporal data. In general, there are different types of temporal data [4]. (1) Sequences: Though there may not be any explicit reference to time, there exists a sort of qualitative temporal relationship between data items. In any transactional data if a transaction appears before another, it implies that the former transaction has occurred before the later. (2)Time Stamped: This category of the temporal data has explicit time related information. (3) Time Series: Time series data is a special case of the time stamped data. In time series data events have uniform distance on the time scale. (4)Fully Temporal: The data of this category is fully time dependent. The inferences are also strictly temporal.

Most temporal data mining techniques transform the temporal data into static representations and exploit existing 'static' machine learning techniques, thus potentially refraining some of the temporal semantics. One major

Downside that arises throughout the mining method is treating data with temporal feature i.e. the attributes connected with the temporal data provided within the database. This temporal attribute requires a different method from other kinds of attributes. However, most of data mining techniques conduce to treat temporal data as an unordered collection of events, ignoring its temporal data [5]. Recently there is a growing interest in the development of temporal data mining techniques in which the temporal dimension is considered more explicitly [6]. Existing approaches to mining temporal association rules rely on identifying all item sets that frequently occur throughout the dataset. With a large number of attributes, this is computationally expensive and can lead to a combinatorial explosion with classical methods.

Association rules that incorporate temporal information have greater descriptive and inferential power and can offer an additional element of interestingness in a number of fields like trading, marketing, social networking, medicos, earth quick detection, robotics and assisted design. A standard temporal association rule is said to be frequent within its Maximum Common exhibition Period (MCP) if and only if its support and confidence are greater than the required minimum support ($MinSup$) threshold and minimum confidence($MinConf$) respectively [7][8]. The problem of mining general temporal association can be decomposed into two steps: (1) Generate all frequent maximal temporal itemsets and the corresponding maximal temporal sub-itemsets with their relative supports; (2) Derive all frequent general temporal association rules that satisfy minimum confidence from these frequent items [9]. In general, temporal association rule mining can be divided into 4 steps as illustrated in Figure 1.

A key issue of classical methods that are based on the support-confidence framework is that temporal patterns having low support values can escape through the minimum support threshold. For example, consider a rule that has high support in the month of December but for the remainder of the year it is relatively much lower. This rule may not be discovered with classical association rule mining algorithms

Abdel Rahman Mahmoud, Dr. Nagy Ramadan, and Abdel Moniem Helmy, "An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 3, pp. 254-261, 2019.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

when there are rules that persistently occur throughout the dataset and consequently have higher support. Assuming that the minimum support is sufficiently low for the rule in December to be discovered, further analysis is required to ascertain any temporal property. One such property, lifespan, was introduced as an extension of the Apriori algorithm [10]. This is a measure of support that is relative to the lifespan of the itemset defined by a time interval, known as temporal support. So, as for the example rule occurring in December, its temporal support would be relevant for a time interval representing December.

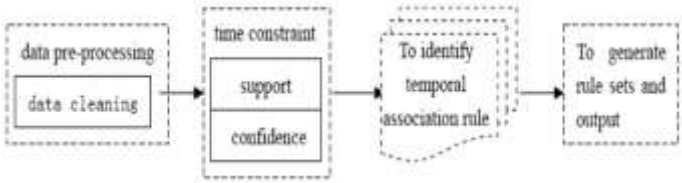

Fig. 1. Steps of temporal association rules mining [4].

Several algorithms have been proposed for mining temporal association rule. These algorithms are based on dividing temporal databases into several partitions according to time and then mining temporal association rules by finding frequent temporal itemsets within these partitions [11]. Progressive Partition Miner (PPM) is proposed as a famous algorithm to discover the general temporal association rule. The basic idea of PPM is to first partition the publication database in the light of exhibition periods of items and then progressively accumulate the occurrence count of each candidate 2-itemset based on the intrinsic partitioning characteristics[12]. The PPM algorithm works efficiently with temporal databases. However, limitation of this technique is its inability to deal with problems of incremental mining and determine the specific partition's size [13]. One extension to PPM model in this paper is to mine general temporal association rules with different start and end times of the transaction set (exhibition periods of the transaction set). This is an interesting yet challenging issue and will be a matter in this research to deal with mining of huge databases.

a. Problem Description

Considering a time-related transaction database $D = \{d_i\}$, each transaction d_i is organized into two attributes values related to the attributes of transaction time and itemset, respectively. Let I be an investigated set of items, and T be an interested time interval (time granularity). Suppose that for each tuple in D, its itemset contains in I and its time falls in T, a temporal association rule in D is an implication of the form x => y in $[t_1, t_2]$, where x, y $\subset I$, x $\cap$ y=Ø, $[t_1, t_2] \subset T$ is the time granularity that x => y is available [14]. The support count of x in $[t_1, t_2]$ is defined as the number of transactions that contain x and occur during $[t_1, t_2]$ in D, and the support of x in $[t_1, t_2]$, support (x, $[t_1, t_2]$), is defined as the ratio of its support count to the number of transactions that occur during $[t_1, t_2]$ in D. A temporal association rule can hold confidence (x => y, $[t_1, t_2]$); the confidence measure can be calculated by support $(x \cup y, [t_1, t_2])$ / support (x, $[t_1, t_2]$). The problem of mining temporal association rules is to find all association rules in the database that satisfies at least three user-specified constraints: minimum support, minimum confidence and time granularity.

The rest of the paper is organized as follows: Section 2 describes some of the recent related works. The detailed description of proposed system has been made in Section 3. In Section 4, the results and discussions on the dataset are given. Finally, conclusions are drawn in Section 5.

II. Related Work.

Several algorithms have been proposed for mining temporal association rule. These algorithms are based on dividing temporal databases into several partitions according to time and then mining temporal association rules by finding frequent temporal itemsets within these partitions [12]. In The PPM algorithm [15-19], the database is first partitioned by the size of time granularity. The PPM algorithm is applied with a filtering threshold mechanism on each partition of the database to prune out those cumulatively infrequent 2-itemsets. The authors in [12] introduced Segment Progressive Filter algorithm (SPF) that first divides the database into certain imposed time granularity. Then in the illumination of the exhibition period of each item, it further segments the database based on their common starting and ending times. For each part of the database, it finds the 2-candidate item set with a cumulative filtering threshold. SPF is also employs scan reduction technique for generating candidate K-item set. After generating all candidates it generates the sub-candidate and counts for the value of support. Limitation of SPF algorithm did not perform any incremental mining technique on the refreshing database [12].

A similar type of approach is adopted in [20] based on the new fast update algorithm (NFUP) that does not require the rescanning of the original database to detect new frequent itemsets. NFUP partition the incremental database by the time interval. The frequent set of itemsets of database is known in advance. NFUP scans each partition backward; the last partition is scanned first. This algorithm in [20] is a modified version of the borders algorithm that minimizes unnecessary candidate generations. The borders algorithm finds the frequent itemsets from the dynamic dataset, using the frequent itemsets already discovered from the old dataset. Here, an infrequent itemset is termed as border set if all the non-empty proper subsets of it are frequent. However, this modified algorithm uses an additional user parameter apart from the parameter support count which is sensitive. With proper tuning of these parameters only a better performance of the algorithm is possible. When this additional parameter's value is closer to the support count; the algorithm converges to the borders algorithm.

Another group of researches in this area is based on the Incremental Standing method for Segment Progressive Filter (ISPF) that modifies the frequent patterns in pace with changes to the database over the time. ISPF first divides the database according to the common start and end times of the itemsets, and it considers the updates of the temporal association rules. ISPF optimized such that scanning of the database is minimized [21]. Methods [22] rely on Recurrent Frequent Pattern Mining concept (RFPM). The RFPM partitions the incremental database logically according to unit

Abdel Rahman Mahmoud, Dr. Nagy Ramadan, and Abdel Moniem Helmy, "An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 3, pp. 254-261, 2019.

time interval (month, quarter or year). The RFPM progressively accumulates the occurrence count of each candidate, according to the partitioning characteristics and works efficiently as compared to previous algorithms.

Another trend to deal with the problem of temporal association rule mining is based on T- Apriori concept [23]. T-Apriori algorithm refers time as a constraint. First of all, we need to analyse the temporal database with respect to time threshold. Time threshold is the time point or time range. Time range can be expressed as $[min\_ts, min\_te]$, while time point $[min\_t]$. In this case temporal association rule, which could be described as $X \rightarrow Y$ (support, confidence, $[ts, te]$).Their example experiment showed that T-Apriori algorithm can successfully extract temporal association rules which described the close relationship between attribute and events. Furthermore, the authors in [24] proposed a temporal association rule mining algorithm for interval frequent-patterns, called $GLFMiner$, which automatically and efficiently generates all intervals without prior domain knowledge in an efficient manner. $GLFMiner$ considers not only global frequent-patterns, but also local frequent-patterns. Experimental results revealed that this algorithm mines more temporal frequent-patterns than traditional association rule mining algorithms and is effective in real-world market basket analysis.

All of the above mentioned approaches assume time partitions are fixed and given beforehand. In other word they rely on an expert to manually specify time granularity. The main contribution is to modify the method of partitioning the database by selecting the biggest partition that contains a large amount of transactions instead of periodically time partitioning, in order to avoid multi-scanning of the database. In addition, the proposed system executes a prior algorithm only to the elected partition aimed to reduce extract time of association rules from the dataset. The value of the proposed temporal miner algorithm for huge database is as follows: (1) The results mined using the system include most of the rules that can be mined by using conventional association rule algorithms;(2) it can discover association rules that are frequent in some intervals, but not throughout the entire dataset; (3) The mined association rules include time attributes that make them more useful; and (4) Valuable rules can still be mined by using larger $MinSup$ thresholds.

III. Proposed Enhanced Temporal Association Rule Algorithm

Let n be the number of partitions with a time granularity, e.g., business-week, month, quarter, year, to name a few, in database D. In the model considered, $db^{t,n}$ denotes the part of the transaction database formed by a continuous region from partition Pt to partition Pn and $\left|db^{t,n}\right| = \sum_{h=t,n}\left|P_h\right|$ where $db^{t,n} \subseteq D$. An item $x^{x.start,n}$ is termed as a temporal item of x, meaning that $P_{x.start,n}$ is the starting partition of x and n is the partition number of the last database partition retrieved. In addition, $\left|db^{t,n}\right|$ is the number of transactions in the partial database $db^{t,n}$ .The fraction of transaction T supporting an itemset X with respect to partial database $db^{t,n}$ is called the support of $X^{t,n}$ i.e., $supp\left(X^{(MCP(X)}\right) = \frac{\left|\left\{T \in db^{MCP(X)} \middle| X \subseteq T\right\}\right|}{\left|db^{MCP(X)}\right|}$. The support of a rule $\left(X \Rightarrow Y\right)^{MCP(XY)}$ is defined as [136]

$$supp\left(\left(X \Rightarrow Y\right)^{MCP(XY)}\right) = supp\left(\left(X \cup Y\right)^{MCP(XY)}\right), \quad (1)$$

The confidence of this rule is defined as:

$$conf\left(\left(X \Rightarrow Y\right)^{MCP(XY)}\right) = \frac{supp\left(\left(X \cup Y\right)^{MCP(XY)}\right)}{supp\left(X^{MCP(XY)}\right)}, \quad (2)$$

A general temporal association rule $\left(X \Rightarrow Y\right)^{MCP(XY)}$ is termed to be frequent if and only if $supp\left(\left(X \Rightarrow Y\right)^{MCP(XY)}\right)$ > min_supp and $conf\left(\left(X \Rightarrow Y\right)^{MCP(XY)}\right)$ > min_conf

Example 1. Consider the database in Figure 2. Since database D records the transaction data from January 2001 to March 2001, database D is intrinsically segmented into three partitions $P1, P2$, and $P3$ in accordance with the "month" granularity. As a consequence, a partial database $db^{2,3} \subseteq D$ consists of partitions $P2$ and$P3$. A temporal item $E2,3$ denotes that the exhibition period of $E2,3$ is from the beginning time of partition P2 to the end time of partition$P3$.

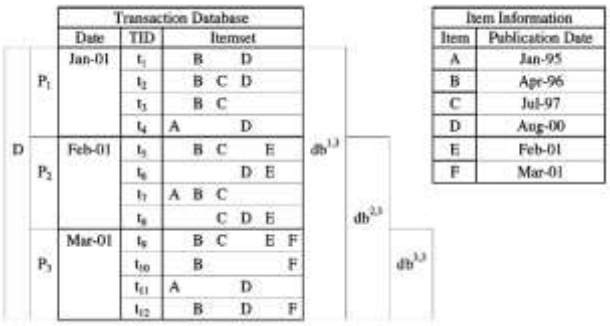

Fig. 2. An illustrative transaction database and the corresponding item information.

In general the mining of general temporal association rule can also be decomposed into three steps: (1) Generate all frequent maximal temporal itemsets (TIs) with their supported values. (2)Generate the supported values of all corresponding temporal sub-itemsets (SIs) of frequent TIs. (3)Generate all temporal association rules that satisfy min_conf using the frequent TIs and/or SIs. Flowing this further, we propose to use a new partitioning approach to address the problem of multi-scanning of huge database which are dynamically adapted depending on indicating partition that contains the biggest amount of transactions with a time granularity. The general framework of the proposed association rule mining system is illustrated in Figure 3, and the following subsections describe each step in details.

Abdel Rahman Mahmoud, Dr. Nagy Ramadan, and Abdel Moniem Helmy, "An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 3, pp. 254-261, 2019.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

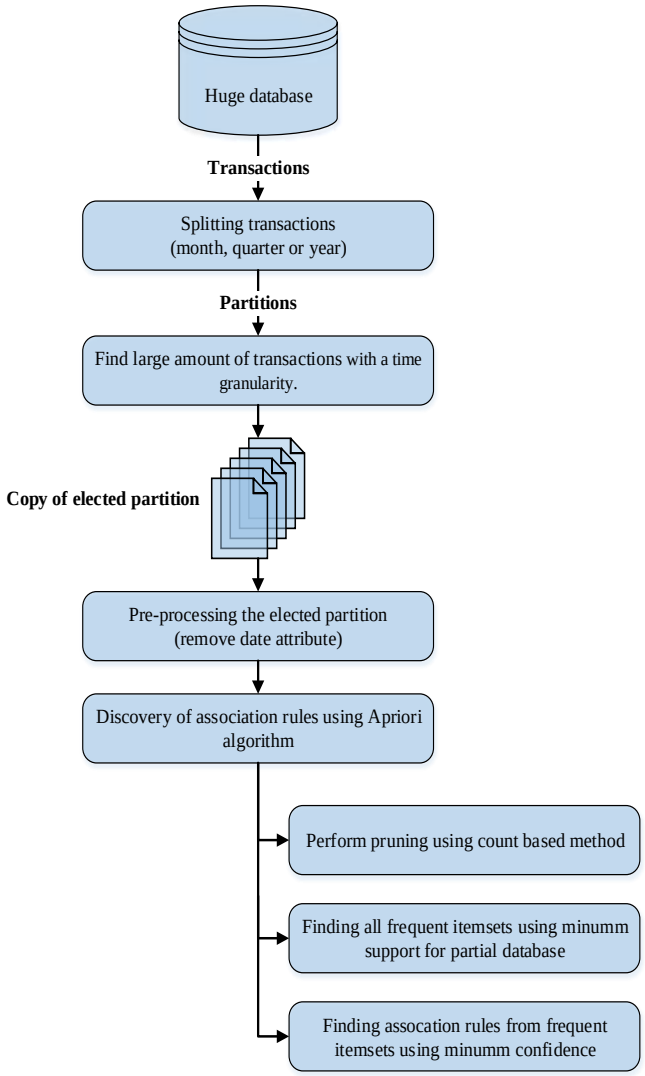

Fig. 3. An Enhanced Temporal Association Rule Algorithm

A. Recall the Database Transactions

In the first stage, the system loads all database transactions T on the memory. Here two types of large database are used syntactic and real database.

B. Splitting Transactions

Generally, scanning the original database is very expensive and time consuming to extract association rules. How to choose a principle to divide the database as a partition *P* through a temporal period of time. This study proposes a new algorithm to partition the database logically according to unit time interval (month, quarter or year).

A temporal situation can be described at different levels of abstraction depending on the accuracy required or the available knowledge. Time granularity can be defined as the resolution power of the temporal qualification of a statement. Providing formalism with the concept of time granularity makes it possible to model time information with respect to differently grained temporal domains. This does not merely mean that one can use different time units, e.g., months and days, to represent time quantities in a unique flat temporal model, but it involves more difficult semantic issues related to the problem of assigning a proper meaning to the association of statements with the different temporal domains of a layered temporal model and of switching from one domain to a coarser/finer one. Such an ability of provide and relating temporal representations at different "grain levels" of the same reality is both an active research theme and a major requirement for many applications (e.g., integration of layered specifications and agent communication).

C. Find the large Amount of Transactions with a Time Granularity.

After the partitioning step, the idea is to select the partition Pe which covers the biggest amount of transactions transaction Di founded in time granularity$T$. In general, a temporal situation can be described at different levels of abstraction depending on the accuracy required or the available knowledge. Time granularity can be defined as the resolution power of the temporal qualification of a statement. Providing formalism with the concept of time granularity makes it possible to model time information with respect to differently grained temporal domains. This does not merely mean that one can use different time units, e.g., months and days, to represent time quantities in a unique flat temporal model, but it involves more difficult semantic issues related to the problem of assigning a proper meaning to the association of statements with the different temporal domains of a layered temporal model and of switching from one domain to a coarser/finer one. Such an ability of providing and relating temporal representations at different "grain levels" of the same reality is both an active research theme and a major requirement for many applications (e.g., integration of layered specifications and agent communication). After that, the selected partition is copied form the original database to anther database (the designated database) for further processing in this set of transactions to find the frequent itemset in this time intervalPe.

D. Re-processing the Elected Partition

To extract the association rules inside the selected database partition Pc, the Apriori algorithm is adapted. However, an Apriori algorithm builds rules in the form of 0 and 1, so the date attributes must be removed from the transactions. This step is automatically applied to Weka software. Weka is a collection of machine learning algorithm for data mining tasks. The algorithm can either be applied directly to a dataset or called from your own Java code. Weka contains tools, data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. In general, the file structure that should be used with A priori method inside Weka program is as follows (*.raff): Header: "@relation <Relation Title> Declaration: Set or group of fields in the structure, where '0' is false and '1' is true value. " @attribute <Filed Title> {0, 1}, and "Data" @data" as illustrated in Fig.4. Fig.5 shows the file structure after eliminating date attribute.

Abdel Rahman Mahmoud, Dr. Nagy Ramadan, and Abdel Moniem Helmy, "An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 3, pp. 254-261, 2019.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

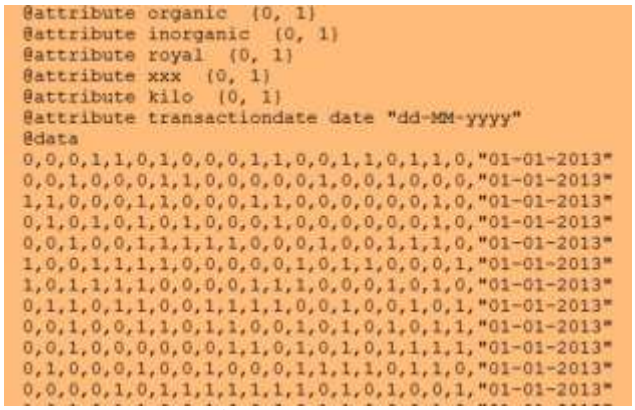

Fig. 4. Weka file's structure with the date attribute.

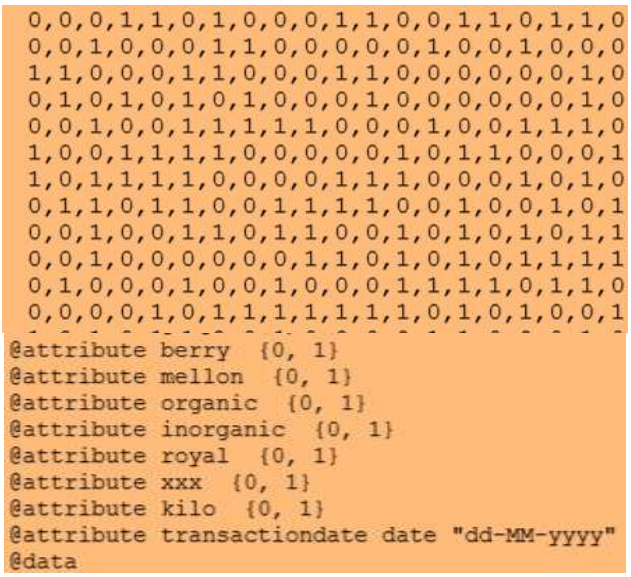

Fig. 5. Weka files structure after removing the date attribute.

E. Discovery of Association Rules using Apriori Algorithm

Apriori employs an iterative approach known as a level wise search, where k-itemsets are used to explore $(k+1)-$itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L_1. Next, L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-temsets can be found. The finding of each L_k requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented is used to reduce the search space. Apriori property: All nonempty subsets of a frequent itemset must also be frequent. A two-step process is used to find the frequent itemsets: join and prune actions.

The key for optimization of the computation of association rules lies in the techniques used to create candidate itemsets. The smaller the number of candidate itemsets is, the faster the algorithm would be. However, in large retailing applications, the number of transactions might be in the order of millions, and the number of items might be in the order of thousands. If there are m items in the database, then there are 2m possible itemsets, indicating that brute force search techniques require exponential time. Thus, many fast algorithms are proposed to find the frequent patterns in a data set working in time linearly to the number of large itemsets and number of transactions (tuples) in the database. These algorithms use clever data structures to speed up the search process [25-27].

IV. Experimental Results

For obtaining reliable experimental results, the method to generate synthetic transactions was employed in this study using C#. Explicitly, the C# code generates several different transaction databases from a set of potentially frequent itemsets to evaluate the performance of the proposed algorithm. These transactions mimic the publication items in a supermarket database. Furthermore, the efficiency of proposed algorithm has been evaluated by some real supermarket databases. However, this section shows the experimental results from synthetic transaction data so as to obtain results of different workload parameters. Each database consists of |D| transactions and, on the average, each transaction has |T| items. To simulate the characteristic of the exhibition period in each item, transaction items are uniformly distributed into database D with a random selection. In accordance with the exhibition periods of items, database D is divided into n partitions. Table 1 summarizes the description of the databases used in the experiments.

Table 1. Dataset description

Dataset	Characteristics
Synthetic database	- 20000 training examples - 20 numerical attributes - 1 MB as uncompressed text
Weka database	- 30000 training examples - 200 numerical attributes - 1.97 MB as uncompressed text
Real database	- 500000 training examples - 20 numerical attributes - 17 MB as uncompressed text

A. Evaluation Criteria

In the experiments, both of Similarity Rate; (S_{rate}) that is defined as the percentage of similar rules out of all tested rules and Dissimilarity Rate (D_{rate}) that is defined as how many dissimilar rules are not corresponding from the total number of rules when demerits the algorithm for all databases.

$$\text{Similarity Rate } (S_{rate}) = \frac{\text{number of Similar rules}}{\text{total number of rules}} \times 100 \quad (1)$$

$$\text{Dissimilarity Rate } (D_{rate}) = 1 - (S_{rate}) \times 100 \quad (2)$$

B. Experimental Results

This section describes in detail the experiments of the proposed algorithm. The synthetic dataset generated from the C# dataset generator and the real datasets were used to validate the effectiveness of the proposed algorithm. The relation between the number of transactions in the selected partition (see Fig. 6) and the similarity rate is determined by the variability of the total number of transactions in the database.

Abdel Rahman Mahmoud, Dr. Nagy Ramadan, and Abdel Moniem Helmy, "An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 3, pp. 254-261, 2019.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Fig. 6. Number of transactions in (P_e).

Table 4 and 2 present the results obtained using the synthetic dataset where min_conf= 50% and the time granularity is varied from day, week, and month. Table 2 presents the outcomes found using the synthetic dataset where min_conf= 50% and the time granularity is wide-ranging from month, three months, and six months. In both experiments, the similarity rate decreased as the total number of transactions in the database increased; furthermore, this rate increased as the time granularity increased from day to month. Lower time granularity (the total number of transactions in the selected parathion decreased) yields a less number of rules, especially when the min_conf is below 0.5 and min_sup = 0.17. Therefore granularity time is one of the main factors that influence the similarity rate. Another important factor is the total number of transactions.

Table 2. The average ratio of rules similarity (S_{rate}) in the synthetic database under day, week, and month granularity respectively.

Total No. of Transactions |D|	(T_{Pe} /T_d) in day granularity	(T_{Pe} /T_d) in week granularity	(T_{Pe} /T_d) in month granularity	(S_{rule}) Average ratio
1000	87 (11.5%)	181 (18.1%)	562 (56%)	60%
3000	194 (6.4%)	347 (11.5%)	827 (27.5%)	40%
5000	206 (4.1%)	366 (7.3%)	1014 (8.4%)	30%

Table 3. The average ratio of rule similarity (S_{rate}) in the synthetic database under month, three months, and six months granularity respectively.

Total No. of Transactions |D|	(T_{Pe} /T_d) in one month granularity	(T_{Pe} /T_d) in three months granularity	(T_{Pe} /T_d) in six months granularity	(S_{rate}) Average ratio
12000	1014 (8.4%)	2106 (17.5%)	3964 (33%)	20%
15000	955 (6.3%)	2349 (15.6%)	4210 (28%)	10%
20000	912 (4.5%)	2108 (10.5%)	3957 (19.7%)	0%

In Table 3, although the proposed algorithm exhibits a similar relationship between the number of transactions in the database and the similarity rate(S_{rate}) of the real dataset to that of the synthetic dataset; it performs well in finding more similar rules when |D| increased. The (S_{rate}) is still encouraged as the |D| increased. One explanation of this result is that the real datasets exhibit the behaviour of the users, unlike the synthetic dataset that lacks the generic behaviour of the user. In general, the data mining concept depends mainly on the explicit and implicit behavers of the users. So, the proposed system yields accurate results with real datasets.

Table 4. the average ratio of rule similarity (S_{rate}) in the real database under month, three months, and six months granularity respectively.

Total No. of Transactions |D|	(T_{Pe} /T_d) in one month granularity	(T_{Pe} /T_d) in three months granularity	(T_{Pe} /T_d) in six months granularity	(S_{rate}) Average ratio
100000	973 (0.9 %)	2492 (2.4%)	4396 (4.5%)	90%
200000	1833 (0.9%)	6230 (3.1%)	7912 (4%)	70%
500000	4275 (.08%)	7811 (4%)	12145 (2.4%)	60%

To confirm that the proposed algorithm performs well in finding similar rules, the following experiment elucidates the relationship between the number of transactions $|D|$ and the similarity rate threshold in real WEKA dataset. For 30000 transactions, under one month time granularity, the system gives 100% similarity because the data contains correlated information. For three and six months, the system gives 90% and 80% respectively. The reason for these lager values is that this dataset is non-huge on contrast of real dataset that contains a huge number of transactions. Fig. 7 summarizes the S_{rate} outcomes for the three datasets. These results conclude for the fact that the similarity rate depends on the selected partition time that is expected to convey the most appropriate rules related to the whole rules extracted from the database.

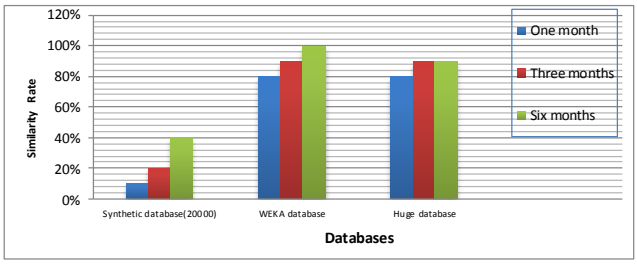

Fig. 7. Comparative results for the proposed approach.

The second experiment examines the scale up performance of the proposed algorithm. The scale-up results for different selected data sets are obtained. The scale-up is the performance of algorithm as the values of $|D|$ increase. The results are obtained for the real database when the number of transactions increases from 100,000 to 500000. Note that the similarity rate only slightly decreases with the growth of the database size, showing the good scalability of the proposed algorithm, meaning that the advantage of the proposed system over traditional algorithm increases as the database size increases.

Abdel Rahman Mahmoud, Dr. Nagy Ramadan, and Abdel Moniem Helmy, "An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 3, pp. 254-261, 2019.

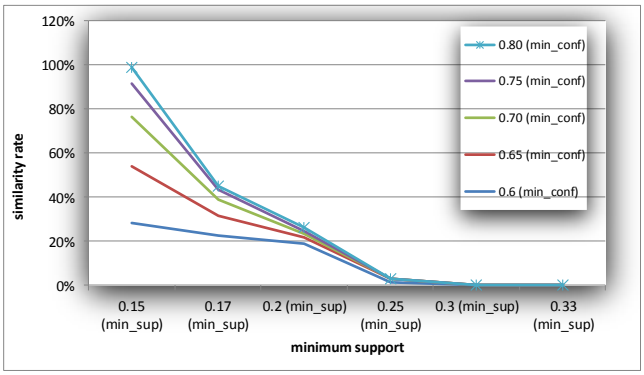

Fig. 8. Relationship between similarity rate (S_{rate}), (min_sup) with different(min_conf).

In the third experiment, the relation between the similarity rate and *min_sup* & *min_conf* settings is determined by varying the *min_sup* thresholds. Fig.8 presents the results obtained using the real 500000 dataset where *min_conf* = 50% to 80% and *min_sup* = 0.15 to 0.33. It can be seen that, lower *min_conf* thresholds yield more similarity rate, with 90 % rate for *min_conf* = 0.8 and *min_ sup* =0.15. This effect is significant since most local frequent-patterns are found with a smaller *min_sup*. The experiments herein reveal that the proposed system finds most potential association rules compared to the original association's rule within selected time interval. In particular, the system resembles traditional approaches that mine in the whole database by a factor of twenty to ninety when *min_sup* is less than 0.15 and 0.2.

V. Conclusion

With the development of technology, the difficulty in obtaining data is decreasing and the difficulty to analyze the huge volume of data is increasing. The value of the proposed algorithm is as follows. The results mined include most of the rules that can be mined by applying the algorithm in the whole database. The system can discover association rules that are frequent in some intervals, but not throughout the entire dataset. The mined association rules include time attributes that make them more useful**.** Valuable rules can still be mined by using larger *min_sup* thresholds. The system automatically generates the intervals of frequent patterns. Finally, the advantage of this approach is that it does not exhaustively search the dataset or require any prior partitioning.

The correctness of the proposed algorithm is proven. Extensive experiments have been performed on synthetic and real datasets to evaluate the performance of the proposed system. Sensitivity analysis of various parameters was conducted to provide many insights into the algorithm varying the value of minimum supports, the size of the time interval, the number of partitions, and the number of transactions. It was noted that the improvement achieved by the proposed system increases as the size of the database increases. In summary, The proposed algorithm looks promising as a method for discovering rules from partition-based data as it reads the database only once for large databases and applies mining in the small subset of the database

Although the proposed approach has been proved for its efficiency, more work needed for the analysis of its effectiveness in terms of quality of rules and its scalability through comparative analysis with other methods. Furthermore, appropriate data structures may be used to further improve the execution time of the proposed system and more real-world datasets will be used to confirm its effectiveness. Finally, optimization techniques like genetic algorithm can be used to select the optimal partitions.

References

[1] Rafi, K., Aiman, M., Hina, K.: ' Data Mining for Marketing', Journal of Marketing and Consumer Research, 2015, 1, (5), pp. 17-28.
[2] Pughazendi, N., Punithavalli M.: 'Discovering Patterns from Temporal Databases Using Temporal Association Rule', Global Journal of Computer Science and Technology, 2011, 11, (14), pp. 1-5.
[3] Kamlesh, M., and Pal P.: ' Mining Frequent Pattern form Large Dynamic Database with Different Exhibition of Time', International Journal of Advanced Research in Computer Science and Software Engineering, 2013,3, (8) , pp. 842- 847.
[4] Mohd, S., Ashish, R., Mohd, D.: ' Temporal Data Mining: An Overview', International Journal of Engineering and Advanced Technology, 2011, 1, (1), pp. 20-24.
[5] Heerash, K., Chandra, P.: 'Improved High Utility Mining Algorithm', International Journal of Scientific and Research Publications, 2013, 3, (10), pp. 1-5.
[6] Jaishree, S., Hari, R., Sodhi, J.: 'Improving Efficiency of Aprioir Algorithm Using Transaction Reduction', International Journal of Scientific and Research Publications, 2013, 3, (1), pp. 1-4.
[7] Sandeep, S., Ganesh, W., Niresh, s.: 'A Review: Data Mining with Fuzzy Association Rule Mining', International Journal of Engineering Research & Technology, 2012, 1, (5), pp. 1-4.
[8] J. Huang, Wei W.: 'Efficient Algorithm for Mining Temporal Association Rule', International Journal of Computer Science and Network Security, 2007, 7, (7), pp. 268- 271.
[9] Thalla, S., Sadak, M.: 'A survey on Temporal High Utility Itemsets on data streams', Engineering Science and Technology: An International Journal, 2013, 3, (6), pp. 753- 756.
[10] Swati, S., Sini, s.: 'Advance Mining of Temporal High Utility Itemset', International Journal of Information Technology and Computer Science, 2012, 4, (4), pp. 26-32
[11] Chang-Hung, L., Cheng-Ru L., Ming-Syan, C.: 'On Mining General Temporal Association Rules in a Publication Database', Proceedings of the IEEE International Conference on Data Mining, Canada, 2001, pp. 337 – 344.
[12] Chang-Hung, L., Cheng-Ru L., Ming-Syan, C.: 'Progressive Partition Miner: An Efficient Algorithm for Mining General Temporal Association Rules', IEEE Transactions on Knowledge and Data Engineering, 2003, 15, (4), pp. 1004- 1017.
[13] Surajit, C., Umeshwar, D., Vivek, N.: 'An Overview of Business Intelligence Technology', Communications of the ACM, 2011, 54, (8), pp. 88-98.
[14] Guojun, M.: 'Mining Temporal Association Rules in Network Traffic Data', International Journal of Future Computer and Communication, 2014, 3, (1), pp. 55-59.
[15] Shruti, A., Ranveer, K.: 'Comparative Study of Various Improved Versions of Apriori Algorithm', International Journal of Engineering Trends and Technology, 2013, 4, (4), pp. 687 – 690.
[16] Darshan, M.: 'Improved Apriori Algorithm for Mining Association Rules', Information Technology and Computer Science, 2014, 6, (7), pp. 15- 23.
[17] Kittipol, W.: 'Association Rule with Frequent Pattern Growth Algorithm for Frequent Item Sets Mining', Journal of Applied Mathematical Sciences, 2014, 8, (98), pp. 4877 – 4885.
[18] MD, J., Shuxiang, X., Byeong, H.: 'Association Rule Mining for Both Frequent and Infrequent Items Using Particle Swarm Optimization Algorithm', International Journal on Computer Science and Engineering, 2014, 6, (7), pp. 221- 231.
[19] Shahana, S., Rupesh, P.: 'A Review on Different Association Rule Mining Algorithms', International Journal of Software & Hardware Research in Engineering, 2014, 3 , (1) , pp. 7- 13.

[20] Monali, P.: 'Incremental Association Rule of Mining for Intrusion Detection', International Journal of Advance Research in Computer Science and Management Studies, 2014, 2, (3), pp. 252-256.
[21] Mohsin, N., Kashif, H., Sohail, A., et al.: 'Role of Segment Progressive Filter in Dynamic Data mining', Journal of Digital Information Management, 2011, 9, (4), pp. 171-175.
[22] Pradnya, S., Vijay, V. 'Mining Frequent Pattern Form Large Dynamic Database with Time Granularities to Improve Efficiency', International Journal of Engineering and Advanced Technology, 2013, 3, (1), pp. 368-372.
[23] Zhai, L., Tang, X., LI, L., et al.: 'Temporal Association Rule Mining Based on T-Apriori algorithm', International Journal Symposium on Spatial Temporal Modeling Analysis, 2005, 5, (2), pp. 1-6.
[24] Brijesh, K., Saurabh, al.: 'Data Mining: A Prediction for Performance Improvement Using Classification', International Journal of Computer Science and Information Security, 2011, 9, (4), pp. 1-5.
[25] Venu, M., Meenu, D.: 'Efficient Association Rule Mining using Hybrid Techniques', International Journal of Electronic and Electrical Engineering, 2013, 6, (2), pp. 77-85.
[26] Nath, B., Bhattacharyya, D., Ghosh, A.: 'Discovering Association Rules from Incremental Datasets', International Journal of Computer Science & Communication, 2010, 1, (2), pp. 433-441.
[27] Anour, F., Ho, S., Khalid, E., et al.: 'IMTAR: Incremental Mining of General Temporal Association Rules', Journal of Information Processing Systems, 2010, 6, (2), pp. 163- 176.

Abdel Rahman Mahmoud, Dr. Nagy Ramadan, and Abdel Moniem Helmy, "An Enhanced Algorithm for Association Rule Mining in Huge Temporal Database," *International Research Journal of Advanced Engineering and Science*, Volume 4, Issue 3, pp. 254-261, 2019.