

NEW TECHNIQUE TO DEAL WITH DYNAMIC DATA MINING IN THE DATABASE

Hebah H. O. Nasereddin

Middle East University,

P.O. Box: 144378, Code 11814, Amman-Jordan

Email: hebah66@hotmail.com

ABSTRACT

Data mining is a part of a process called KDD-Knowledge Discovery in Databases. This process consists basically of steps that are performed before carrying out data mining, such as data selection, data cleaning, pre-processing, and data transformation [1, 2]. There may be thousands or millions of records that have to be read and to extract the rules for, but the question is what will happen if there is new data, or there is a need to modify or delete some or all the existing set of data during the process of data mining. Also real-world databases are highly susceptible to noise, missing, and inconsistent data due to their typically huge size, often several gigabytes or more. The questions here are; how can the data be preprocessed in order to help improve the quality of the data, and consequently the mining results? How can the new updated data be preprocessed in order to help improve the quality of the data, efficiency, and simplify of the mining process? Therefore the purpose of this study is to find solutions (Item-summation) for dynamic data mining process that is able to take into considerations all updates (insert, update, and delete problems) into account.

Key words: *Static, data mining process, Item-summation.*

1. INTRODUCTION

Data mining is the task of discovering interesting and hidden patterns from large amounts of data where the data can be stored in databases, data warehouses, OLAP (on line analytical process) or other repository information. It is also defined as Knowledge Discovery in Databases (KDD) [3].

Data mining is one of the most important research fields that are due to the expansion of both computer hardware and software technologies, which has imposed organizations to depend heavily on these technologies [4]. Data is considered as the number one asset of any organization, it is obvious that this asset should be used to predict future decisions [5]. Consequently, and since organizations are continuously growing, their relative databases will grow as well. The organizations current data mining techniques as a result will fail to cope up with large databases which are dynamic by nature [6]. Data mining is the way to help organization make full use of the data stored in their databases [7], it is the best way for all fields and for all different types of organizations especially when it comes to decision making.

Databases tend to be large and dynamic thus their contents usually do change; new information might need to be inserted, current data might need to be updated and/or deleted. The problem with this from the data mining perspective is how to ensure that the rules are up-to-date and consistent with the most current information. Also the learning system has to be time-sensitive as some data values vary over time and the discovery system is affected by the correctness of the data.

Once a data mining system is installed and is being used in daily operations, the user has to be concerned with the system's future performance because the extracted knowledge is based on past behavior of the analyzed objects [2]. So:

- If future performance is very similar to past performance (e.g. if company customers files do not change their files over time) using the initial data mining system could be justified.
- If, however, performance changes over time (e.g. if hospital patients do not change their files over time), the continued use of the early system could lead to an unsuitable results and (as an effect) to an unacceptable decisions based on these results [8].

Here is where dynamic data mining comes into play by offering logical suitable techniques for “updating”. In practice, and looking to empirical cases, dynamic data mining could be extremely helpful in making the right decision in the right time and affects the efficiency of the decision as well [9].

In this respect and in view of what have been introduced regarding dynamic data mining and its importance and its effects on decision making. It is our intention to put forward a solution in order to run data mining without the need to restart the whole process every time there are changes on the data being used, in other words the running process should focus solely on the amendments taking into consideration that the mining run is held constant.

2. STATIC DATA MINING PROCESS

Data mining process is a step in Knowledge Discovery Process consisting of methods that produce useful patterns or models from the data [10]. Some problems might occur because of duplicate, missing, incorrect, outliers’ values, and sometimes a need to make some statistical methods might arise as well, even though when the problem was known, and correct data is available as well.

The KDD procedures are shown below in a way to help us focus on data mining process. It includes five processes: 1) Defining the data mining problem, 2) Collecting the data mining data, 3) Detecting and correcting the data, 4) Estimating and building the model, 5) Model description and validation, as seen in Figure.1 [12].

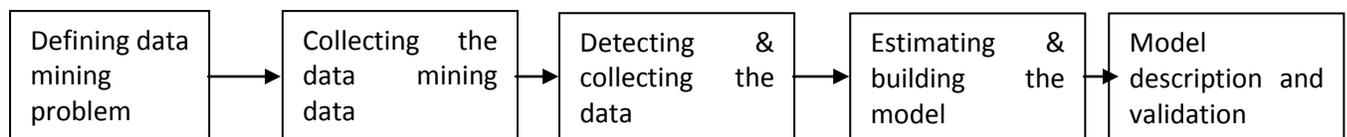


Figure 1: Data mining process

3. DYNAMIC DATA MINING PROCESS

As mentioned earlier many researchers and developers have specified a process model designed to guide the user through a sequence of steps that will lead to good results. Many have been reported for data mining process. Some of these assumed that this is possible for Dynamic data mining process. Up-to-date most of the data mining projects have been dealing with verifying the actual data mining concepts. Since this has now been established most researchers will move into solving some of the problems that stand in the way of data mining, this research will deal with such a problem, in this case the research is to concentrate on solving the problem of using data mining dynamic databases.

3.1 DYNAMIC ITEM-SUMMATION

The designers of the data mining process; in most cases, probably do not know much about the data sources; if they do, they mostly wouldn't like to be interested in performing data mining or dynamic mining. Individually, the data seem simple, complete, and explainable. But collectively, they take on a whole new appearance that is intimidating and difficult to comprehend. Real-world databases are highly susceptible to noise, missing, and inconsistent data due to their typically huge size, often several gigabytes or more. The questions here are; how can the data be preprocessed in order to help improve the quality of the data, and consequently the mining results? How can the new updated data be preprocessed in order to help improve the quality of the data, efficiency, and simplify of the mining process? Also now, all the modifications of dynamic data mining were for insert case but not on update or delete case. From this, the importance of the new technique appears and for this reason this problem is going to be the main topic of this paper.

3.2 Basic Terminology

This paper proposes a new efficient algorithm that will detect and selects records that have been changed and/or modified after they have been collected and/or used in a mining run, regardless of the size of data available within the source database. This algorithm generates a mathematical summation for each record. Based on these summation values the exact record in the local database that have been modified and needs to be replaced can be identified. In other words, if there are any modification, and/or deletion (affecting one or a number of records); the application must not read and/or select all the records; in order to identify the modified record. It simply selects the records summation; for the particular record, and make the changes needed, related to the record with the modified summation value, this will result in the replacement of the records by their changed value from the source DB.

In this study it is assumed that we have already dealt with the data selection process, the required data are already available in the local DB. A number of processes have already been carried out on the data including the data cleaning process.

3.3 Global View of the Proposed Algorithm

Before presenting the technique used in this paper, it is important to mention that this technique is applied during a data mining run, and after the data cleaning sub process.

The proposed algorithm follows the steps listed below

Step I:

Read selected data from local database after cleaning the data and during the data preprocessing operation.

Step II:

Create a Table. The rows represent the number of records while columns represent the number of attributes. This Table will be filled with numbers, [as shown in step (III)]

Step III:

The parameters for each nominal attribute are read without duplication. During this process the parameters are numbered starting from zero (0 - n-1); in the order they appeared in the list. This process decreases the number of characters; which will in turn decrease the data size, this is due to the decreased number of calculations needed in the algorithm, and this number will be treated as a special number. For example if it is required to examine the data say the gender; if the parameters for the gender without duplication, it should return only {male, female}; which can be assign (0) for males and (1) for females. Numbering the parameters is for nominal attributes only; in the case where the attribute type appears as integer or double the parameters are kept as they are. In the case where the parameters are missing the parameter are changed to the Symbol (?).

Step IV:

Create a second Table with #of attributes +1

The rows represent number of records while columns represent the number of attributes. Each number that represents a parameter in the first Table will be transformed to a new number in the second Table. The proposed algorithm will add a new column to the new Table which represents the summation of the calculated numbers, taking into consideration the position of the number in the Table. The calculations for each cell in the new Table are carried out as follows:

- a- If the number is zero, it is changed to the number of column.
- b- If the parameter is (?) "Missing value", it is changed to the negative number equivalent to its column numbers (index).
- c- If the parameter is not (?) and/or zeros, it is calculated as: The number representing the parameter is converted to its binary equivalent; this is then converted to a number according to equation :

$$\text{calculated_parameter}(i) = \sum_{i=1}^n d_i * B^i$$

Where:

- N*: is the number of digits in binary representations,
d_i: is the *i*th digits
B: is the column number+1

For example: suppose the number that represents the parameter in column 2 is (11)₁₀, its binary equivalent is (1011)₂, using the equation, will produce the summation as:

$$\text{calculated_parameter} = 1 \times 3^1 + 1 \times 3^2 + 0 \times 3^3 + 1 \times 3^4 = 93$$

- d- The result of the above summation represents the parameters in each column which will appear in the Table added column.

A simple example containing small numbers, the summation for the two rows equal three, both have one question mark, one #1, one #2, two # zero but in different places :

Table1: an example before using Item-summation

1	2	0	?	0	3
0	2	?	0	1	3

The first row will be:

$$(1)_{10} = (001)_2, \text{calculated_parameter}(1) = 1 \times 2^1 = 2$$

$$(2)_{10} = (010)_2, \text{calculated_parameter}(2) = 0 \times 3^1 + 1 \times 3^2 = 9$$

0: replaced by column # = 3

? :replaced by column # = -4

0: replaced by column # = 5

The summation value for row # 1 will be equaled 15

The second row will be:

0: replaced by column # = 1

$$(2)_{10} = (010)_2, \text{calculated_parameter}(2) = 0 \times 3^1 + 1 \times 3^2 = 9$$

? :replaced by column # = -3

0: replaced by column # = 4

$$(1)_{10} = (001)_2, \text{calculated_parameter}(5) = 1 \times 6^1 = 6$$

The equivalent summation value for row # 2 will be equaled 17

The final result will be:

Table2: the example after using Item-summation

2	9	3	-4	5	15
1	9	-3	4	6	17

The summation that appears in the added column will help the algorithm in identifying and selecting any modified records, so as to make the necessary changes. In the case where changes and/or modification on the selected data have taken place, it is dealt with in the following manner:

- If there are new records, just insert the record to the Table after making the necessary calculations.
- If there is an update on a record, and/or records, find the summation related to that record, delete (remove) the old record from the Table, insert the updated record in its place, this should be followed by carrying out the necessary calculations of the new summation value, which should replace the old summation value.
- If the operation is to delete records, find the summation related to the deleted records and delete them from the Table.

3.4 Detailed Description of the Proposed Algorithm

The following example will illustrate in great details how the application will deal with the different cases mentioned above, as and when they take place. It will also, show how the steps mentioned above will be carried out. Assuming that in this example 15 records are chosen randomly with 9 attributes selected from the main database, the steps carried out by the algorithm are:

Step I: Reading the selected data from local database.

The selected items (name of attributes) which are read from local database are:

MARTIAL_STATUS,BLOOD_TYP,REFLERDBY,CONTRY,AGE_YR,BASIC_SALARY,WORK_CITY,SOCIALNUMBER,GENDER

The data is as follows:

MARRIED,O+,ROYALMED,JORDAN,30,59,IRBID,9732013876,FEMALE,
 DEVORCED,O-,null,JORDAN,75,null,null,9311007008,MALE,
 MARRIED,O-,ROYALMED,JORDAN,62,null,null,9391011760,MALE,
 MARRIED,A-,WITHOUT,JORDAN,33,null,null,9682038307,FEMALE,
 MARRIED,O+,WITHOUT,IRAQ,24,null,null,2000426745,FEMALE,
 MARRIED,O+,EMERGENCY,JORDAN,49,null,null,9562016841,FEMALE,

MARRIED,O+,PRIVE.SECTOR,JORDAN,70,null,null,9352011581,FEMALE,
 SINGLE,O+,GOV.CENTER,JORDAN,24,45,JARASH,9811013798,FEMALE,
 SINGLE,O+,GOV.CENTER,JORDAN,38,67,TAFELEH,9671027564,MALE,
 MARRIED,O+,WITHOUT,PALISTINE,23,null,null,9761014305,FEMALE,
 SINGLE,O+,WITHOUT,EGEPT,31,null,null,1104101430,FEMALE,
 DEVORCED,O+,null,JORDAN,20,null,null,9812000963,FEMALE,
 WIDOW,O+,null,JORDAN,71,null,null,9351006836,MALE,
 WIDOW,O+,WITHOUT,JORDAN,null,null,null,9351006836,MALE,
 MARRIED,O+,OTHER,JORDAN,20,null,null,9812057684,FEMALE,

Step II: A Table is created for the above data, this is represented as:

SS_MARTIAL_STATUS	SS_BLOOD_TYP	I_BASIC_SALARY	I_WORK_CITY	HOSP_CONTR	HOSP_AGE_YR	HOSP_REFLEDBY	SOCIALNUMBER	SS_GENDER
MARRIED	O-			JORDAN	62	ROYALMED	9391011760	MALE
MARRIED	A-			JORDAN	33	WITHOUT	9682038307	FEMALE
MARRIED	O+			IRAQ	24	WITHOUT	2000426745	FEMALE
MARRIED	O+			JORDAN	49	EMERGENCY	9562016841	FEMALE
MARRIED	O+			JORDAN	70	PRIVE.SECTOR	9352011581	FEMALE
SINGLE	O+	45	JARASH	JORDAN	24	GOV.CENTER	9811013798	FEMALE
SINGLE	O+	67	TAFELEH	JORDAN	38	GOV.CENTER	9671027564	MALE
MARRIED	O+			PALISTINE	23	WITHOUT	9761014305	FEMALE
SINGLE	O+			EGEPT	31	WITHOUT	1104101430	FEMALE
DEVORCED	O+			JORDAN	20		9812000963	FEMALE
WIDOW	O+			JORDAN	71		9351006836	MALE
WIDOW	O+			JORDAN		WITHOUT	9351006836	MALE
MARRIED	O+			JORDAN	20	OTHER	9812057684	FEMALE
MARRIED	O+	59	IRBID	JORDAN	30	ROYALMED	9732013876	FEMALE
DEVORCED	O-			JORDAN	75		9311007008	MALE

Figure 2: The new table.

Step III: the parameters for each nominal attribute are read without duplication. During this process the parameters are numbered starting with zero in the order they appeared in the list.

The selected nominal attribute without duplication are:

List Parameters

- (1) SS_MARTIAL_STATUS {DEVORCED MARRIED SINGLE WIDOW}
- (2) SS_BLOOD_TYP { A- O+ O- }
- (3) HOSP_REFLEDBY {EMERGENCY GOV.CENTER GOV.HOSPITAL OTHER PRIVE.SECTOR ROYALMED WITHOUT}
- (4) HOSP_CONTRY {ALGERIYA EGEPT EMAREITS IRAQ JORDAN LEBNON MURITANYA OMAN PALISTINE SUDAN SYRIA YOGSLAFIA}
- (5) HOSP_AGE_YR
- (6) I_BASIC_SALARY
- (7) I_WORK_CITY {IRBID JARASH OTHERS TAFELEH ZARQA}
- (8) SOCIALNUMBER
- (9) SS_GENDER {FEMALE MALE}

Note: HOSP_AGE_YR and SOCIALNUMBER are integers and there for they are kept as they are, as shown bellow:

In this case if we assume

- (1) SS_MARTIAL_STATUS {DEVORCED MARRIED SINGLE WIDOW}

Number zero was assigned to DEVORCED, 1 to MARRIED, 2 to SINGLE, 3 to WIDOW.

In the case where the attribute type appeared as integer or double the parameters are kept as they are. For example

MARRIED,O+,ROYALMED,JORDAN,30,59,IRBID,FEMALE,

This is converted to

1.0 1.0 5.0 4.0 30.0 59.0 0.0 0.0

In the case when a parameter is a “missing value” the parameter is replaced by the Symbol (?). For instance if

MARRIED,O+,OTHER,JORDAN,20,null,null,FEMALE,

This is converted to

1.0 1.0 3.0 4.0 20.0 ? ? 0.0

Reading the selected data from local database will produce the information in report

LIST Descriptions

- (1) nominal: SS_MARTIAL_STATUS { DEVORCED MARRIED SINGLE WIDOW }
- (2) nominal: SS_BLOOD_TYP { A- O+ O- }
- (3) int: I_BASIC_SALARY
- (4) nominal: I_WORK_CITY { IRBID JARASH TAFELEH }
- (5) nominal: HOSP_CONTRY { EGEPT IRAQ JORDAN PALISTINE }
- (6) int: HOSP_AGE_YR
- (7) nominal: HOSP_REFLERDBY { EMERGENCY GOV.CENTER OTHER PRIVE.SECTOR ROYALMED WITHOUT }
- (8) int: SOCIALNUMBER
- (9) nominal: SS_GENDAR { FEMALE MALE }

READING FILE:

inputFileName = C:\Documents and Settings\dell 2006\My Documents\tabledata

Number of records = 15

Number of columns = 9

Num. missing values = 28

LIST DATA

```

-----
1.0 2.0 ? ? 2.0 62.0 4.0 ? 1.0
1.0 0.0 ? ? 2.0 33.0 5.0 ? 0.0
1.0 1.0 ? ? 1.0 24.0 5.0 ? 0.0
1.0 1.0 ? ? 2.0 49.0 0.0 ? 0.0
1.0 1.0 ? ? 2.0 70.0 3.0 ? 0.0
2.0 1.0 45.0 1.0 2.0 24.0 1.0 ? 0.0
2.0 1.0 67.0 2.0 2.0 38.0 1.0 ? 1.0
1.0 1.0 ? ? 3.0 23.0 5.0 ? 0.0
2.0 1.0 ? ? 0.0 31.0 5.0 ? 0.0
0.0 1.0 ? ? 2.0 20.0 ? ? 0.0
3.0 1.0 ? ? 2.0 71.0 ? ? 1.0
3.0 1.0 ? ? 2.0 ? 5.0 ? 1.0
1.0 1.0 ? ? 2.0 20.0 2.0 ? 0.0
1.0 1.0 59.0 0.0 2.0 30.0 4.0 ? 0.0
0.0 2.0 ? ? 2.0 75.0 ? ? 1.0

```

The above data is transformed; the transformed data are of the form given in table (3).

Table 3: The transformed data.

1.00	2.00	?	?	2.00	62.00	4.00	?	1.00
1.00	0.00	?	?	2.00	33.00	5.00	?	0.00
1.00	1.00	?	?	1.00	24.00	5.00	?	0.00
1.00	1.00	?	?	2.00	49.00	0.00	?	0.00
1.00	1.00	?	?	2.00	70.00	3.00	?	0.00
2.00	1.00	45.00	1.00	2.00	24.00	1.00	?	0.00
2.00	1.00	67.00	2.00	2.00	38.00	1.00	?	1.00
1.00	1.00	?	?	3.00	23.00	5.00	?	0.00
2.00	1.00	?	?	0.00	31.00	5.00	?	0.00
0.00	1.00	?	?	2.00	20.00	?	?	0.00
3.00	1.00	?	?	2.00	71.00	?	?	1.00
3.00	1.00	?	?	2.00	?	5.00	?	1.00
1.00	1.00	?	?	2.00	20.00	2.00	?	0.00
1.00	1.00	59.00	0.00	2.00	30.00	4.00	?	0.00
0.00	2.00	?	?	2.00	75.00	?	?	1.00

Step IV: The second table is created; each number that represents a parameter in the first table is transformed to a new number in the second table. The data in Table1 (table1) above is transformed to Table2.

Figure 3: The transformed data from Table1 to Table2.

The proposed algorithm adds a new column to the new Table; this is used for the calculated summation value.

Figure 4: The data in Table2 with the summation column.

As a conclusion figure 2 and figure 4 shows the comparison between the original data format and the final data format in (Table2). Note that the final format of Table2 (figure 4) contains only numeric data type.

3.4.1. Proposed Algorithm.

The following notations are used:

C: number of attributes.

C_i : attribute number i.

R: number of records.

R_i : record number i.

Table: data selected from local database.

Table1: table contains (R, C).

Table2: table contains (R, C+1).

Table list [I, N]: contain list of nominal for C_i without duplicate.

N_i : number without duplication for C_i .

Itemsum[I]: the summation of C_i for Table2

Input: select data from local database after cleansing

Output: a summation column for each record

Begin

```

1: Create Table1, Create Table list [I, number],
   Read Table
2: For I = 1 to C Do      loop1
2.1:   N [I] = 0, Table list [I, N] = NULL
2.2   For J = 1 to R Do      loop2
2.2.1  If Table [I, J] = null Table1 [I, J] = '?'; break
2.2.2  If  $C_j$  type = integer or double
       Table1 [I, J] = Table [I, J]
       Else  $C_j$  type = nominal
       If Table [I, J] exists in Table list
       Table1 [I, J] = Table list [I, M]
       Else Table1 [I, J] = N, N [I] = N [I] + 1
2.3   end loop2, end loop1
3: Create Table2
4: For I = 1 to R Do      loop1
4.1:   Itemsum = 0
4.2:   For J = 1 to C Do      loop2
       4.2.1: Read Table1 [I, J], test = ATable1 [I, J]
       4.2.2: If test = 0, Table2 [I, J] = J
       4.2.3: If test = ? , Table2 [I, J] = (-1)*J
       4.2.4: If test ≠ (0 or "?") Then Do Calc algorithm
       4.2.5: Itemsum = Itemsum + Table2 [I, J]
       4.2.6: End loop2
4.3: Table2 [I, J+1] = Itemsum, End loop1
5:   End

```

Algorithm: Calc algorithm

Input: Table1 [I, J], which its value not (?) or zero

Output: a special number for Table2 [I, J]

Note: this algorithm has special calculations in case the number is not zero or (?) symbol

Begin

```

1: read Table1 [I, J]
2: test1 = the value of converting (Table1 [I, J]) to binary code
3: Edit No = 0; Counter = 0;
4: For (Loop = test1.length - 1; Loop >= 0; Loop--)      loop1
4.1:   Counter = Counter + 1
4.2:   If (test1.charAt (Loop) == '1')

```

```

          Edit No = Edit No + (j+1)Counter
4.3:   Table2 [I, j] = Edit No;
4.4:   end loop1
5:     return Table2 [I, j]
6:     end

```

This data also includes the column summation, which is used to identify the attributes. The summation value will be used in this case to track all changes and/or modifications that might take place in the original DB, if and when any modification takes place the value of the modified attributes will be replaced by the modified value, and a new summation value will be calculated and inserted for it. If the attribute is deleted in the original DB source its equivalent value will be deleted along with its summation value.

4. CONCLUSION

This paper presented a part solution to the problem of Dynamic data mining. It is concerned with the process of detecting an update on the data after it has been collected for the data mining from its original source. The proposed algorithm is able to work massive real-world databases regardless of the amount of data and/or the amount of memory available.

The proposed algorithm uses a mathematical equation; special devised for the process of calculating a summation value. The summation value is unique for each record in the database. It is assumed that if the record is changed this summation value will change accordingly. This algorithm also copies all updates that have taken place in the original database to a dummy table specially created for this purpose. This dummy table will contain a copy of the update records plus their summation value. Records within this table will be deleted as soon as the algorithm deals with them. Based on the summation value all the updated records are identified and all the necessary updates (insert, update, and delete) are carried out on the data used in the data mining process.

There is still much research to be done to enhance the applicability and efficiency of the method introduced in this paper. Below is an enhancement that can be introduced to the developed algorithm.

The Dynamic approach will be tested with different datasets that cover a large spectrum of different data mining applications including distributed databases, such as, web site access in e-commerce advertising.

REFERENCES

- [1]. Daniel T. Larose. *Discovering knowledge in data: an introduction to data mining*, John Wiley & Sons, Hoboken, New Jersey, 2005.
- [2]. Fernando Crespoa and Richard Weberb. "A methodology for dynamic data mining based on fuzzy clustering", *Elsevier, Fuzzy Sets and Systems* 2005 pp 267–284.
- [3]. Fayyad, U. M., G. P. Shapiro, P. Smyth. "From Data Mining to Knowledge Discovery in Databases", *AI Magazine*, American Association for Artificial Intelligence Fall 1996: pp 37–53.
- [4]. R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets of items in large database", *The ACM SIGMOD Conference*, Washington DC, USA, 1993, pp. 207-216.
- [5]. T. Fukuda, Y. Morimoto, S. Morishita and T. Tokuyama, "Mining optimized association rules for numeric attributes," *The ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1996, pp. 182-191.
- [6]. P. Domingos and G. Hulten. "Mining high-speed data streams". *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, August 2000, pp 71–80.
- [7]. R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," *The International Conference on Very Large Data Bases*, 1994, pp. 487-499.
- [8]. David Hand, Heikki Mannila, Padhraic Smyth. *Principles of Data Mining*, MIT Press, Cambridge, MA, 2001.
- [9]. M. El-Hajj and O. R. Za'iane. "Non Recursive Generation of Frequent K-itemsets from Frequent Pattern Tree", In *Proc. of 5th International Conference on Data Warehousing and Knowledge Discovery, DaWak'*, 2003, pp 371–380.
- [10]. Mehmed Kantardzic. *Data Mining: Concepts and Techniques*, John Wiley & Sons, IEEE Press 2003.
- [11]. Hebah H. O. Nasereddin, *Dynamic Data Mining Process, ICITST-2008*, Dublin, Ireland.
- [12]. Hebah H. O. Nasereddin, "Stream Data Mining", *International Journal of Web Applications*, Volume 3, Number 2, June 2011, pp 90- 97.