Intelligent decision support system: Application on university

Ali T.Radeef*, [a] [b] , Murtadha M **, [b]

[a]Maaref University College, Ramadi, Iraq

and

[b]Department of Computer Sciences, College of Computer Sciences and Information Technology, University of Anbar, Ramadi, Iraq

corresponding author: * ali_tr_2009@yahoo.com

**dr.mortadha61@gmail.com

Abstract: Over the last decade, the data warehouse contains large amounts of information, often combine from a variety of independent sources. Decision support functions in a warehouse such as on-line analytical processing (OLAP) and analysis, involve hundreds or thousands of complex aggregate queries over large volumes of data. Warehouse applications, therefore build a large number of summary tables or materialized aggregate views (stored in disk) to help improve system performance. It is not suitable to execution queries by scanning the data sets each time.The information stored at the data warehouse is in forms of the views referred to as materialized views. The design of data warehouse is one of the core research problems in studying and evolution of data warehouse. One of the most important decisions in the design of data warehouse is the data warehouse selected view. The selected views to materialize effects on efficiency as well as the total cost of establishing and running a data warehouse. The main goal of this paper is aggregated data from a different source and consolidate the data imported and summarize this data to avoid analyzing all the dataset. Which helps to reduce implementation time by using two techniques to storage summaries (materialized and index views) which helps to increase query speed and application the hybrid intelligent technology (ANN and decision tree algorithm) for access to the decision support system. The proposed system is applied to graduate students where the percentage of graduate students was determined by department, gender, and age. the proposed artificial neural network and Decision tree algorithm were obtained high efficiency to process large data warehouse size with (7488052 records) at a very little time that is (4 minute and 52 seconds).

Keywords: Data Warehouse, Materialized Views, On-Line Analytical Processing, Artificial Neural Network, Multiple View Processing Plan, Decision tree.

1. Introduction

A Data warehouse (DW) is outlined as an electronic information service that used for storing information that is collected from multiple operational databases so as to modify complicated business analysis queries like account aggregates etc. data warehouse generally used to provide analytical results from multidimensional information through creating account and process of supply information that has relevancy to sure analysis. The aim of victimization information warehouse is to breed new information across the relevant tables and views as quickly as doable [1].

The fundamental definition of data warehouse was given by Inmon [2] that data warehouse could be a subject-oriented, integrated, and time-varying and non-volatilisable assortment of knowledge in support of the management's decision-making method.

A data warehouse is seen as a group of materialized views (MVs) outlined over a group of base relations. On-Line Analytical method (OLAP) and decision support system (DSS) applications used complicated aggregation queries so as to form a call and respondent the question. To urge high question performance, the queries area unit is dead regionally at the data warehouse, victimization the (MVs). Once the bottom relations modification, the materialized views at the data warehouse views has to be updated [1].

A data warehouse is dynamic entities which will be evolving over time. Once time passes, and once the data updated by victimization one among update operations like adding a new row(s), deleting row(s) or change information(s), the data warehouse should be synch nonuse with the change and replicate the changes thereon. The data warehouse will be seen as a group of materialized views outlined over some relations; Queries ought to be answered by these materialized views. a number of the new queries will be answered by the views already materialized within the data warehouse. However alternative new queries ought to materialization new views. so as to a question be answered by the data warehouse, there should exist entire editing of it over the materialized views. Such editing will be over the previous views or over the new views, or part over the new and part over the previous views. Once new views ought to be materialized, a further house has to be allotted for materialization so as to answer the new queries [3].

The decision tree classification technique is easily understandable and accurate as well. This technique consists of a set of algorithms which serve the process to classify the data set and select the set of the most relevant features of the dataset. Basically, the model is based upon feature selection algorithm so that diagnosis is uniform, intelligent and quick[14].

Artificial neural network (ANN) is a mathematical computational model that imitates the biolog-ical neuron capability. One of the main features of neural network is it be able to learn a pattern and apply the "knowledge" to the similar pattern. This capability ena-ble neural network to be used to solve wide range of problems including forecasting and classification problem. In the application of reservoir operation and management, ANN has been applied for various simulation and optimization problem[12].

2. PROBLEM DESCRIPTION

Data warehouse contains an oversized variety of tables that square measure coupled with one another in relationship use (star schema or snowflake schema). Every table contains an oversized variety of columns and rows up into millions. This huge quantity of knowledge and victimization hybrid intelligent decision support systems helps to create a choice. Merge two or a lot of algorithms to create hybrid intelligent call support systems (HIDSS) which may deal a lot of powerfully with problems like fast-learning, uncertainty, on-lineability, information capability, and hierarchical answer, etc.

3. LITERATURE REVIEW

In this sub point we are cover the various research outcome from previous research paper and reports. From this study, we have highlighted the design of knowledge and dynamic data warehouse research. For this study we have studied the following research evidence as papers:

In 2011 Wan Hussain et.al The foretelling model utilize ANN to perform foretelling of the reservoir water level, whereas, within the call model, ANN is applied to perform classification of this and changes of reservoir water level. The simulations have shown that the performances of ANN for each foretelling and call models square measure so-so good [12].

In 2014, GarimaSahu et.al The feature choice strategies area unit applied to extend the quickness of the model. With the assistance of feature choice strategies, all the redundant and unwanted options can get removed and a collection of effective options can solely be needed for the aim of a designation of sickness. Best 1st search and rank search area unit the foremost appropriate feature choice methodology which may be applied to strengthen the potency of the projected model for corium diseases [13].

In 2016 M.Hamad, and Y.Turky The most goals of this work area unit to point out the use of derived knowledge like materialized views for run time re-optimization of combination queries (quick response time), effective, transparency and accuracy area unit vital factors within the success of any knowledge warehouse [5].

In 2017 Yangui et.al Knowledge becomes more and more necessary for several corporations sever day operations. However, thanks to their scale, complexness, and dynamics, these networks area unit tough to be processed by suggests that of ancient warehouse systems. In fact, the ascent of knowledge with the frequent arrival of recent desires needs that the system ought to be flexible to changes. This work presents DW4SN (Data Warehouse for Social Network) tool for building knowledge warehouse from a social network, wherever cluster ways area unit wont to outline the data warehouse schema and NoSQL systems area unit wont to implement the warehouse. The authors tend to validate the system on a true knowledge set regarding crafts girls social network. Most edges were obtained in terms of measurability and dynamicity [11].

4. The Major Phases of The Proposed System

In this section, we have a tendency to list and demonstrate the most steps (phases) of our planned IDSS system. The system phases are explained within the next sections of this chapter very well with examples and algorithms. The results of our IDSS system likewise because the results of the most phases and applied algorithms in these phases are conferred and delineate in chapter four. Figure (1) shows a flowchart illustrating the phases (steps) and sub-steps of the planned IDSS system in additional detail. The foremost phases of the system are:

1. ETL (extract, rework and load) Collect knowledge from completely different sources, these sources is completely different files like (Excel, Access, Word, Text files… etc.) by victimization ETL, the knowledge load to knowledge warehouse is data integration.
2. Create Multi-star schemas
3. Generation set OLAP queries by operation aggregation (sum, max, min, count… etc.)
4. Create materialized read to storage the outline of data into materialized views.
5. Using 1st and 2nd formula of the artificial neural network and decision trees access to decision support systems.
6. decision support system result.

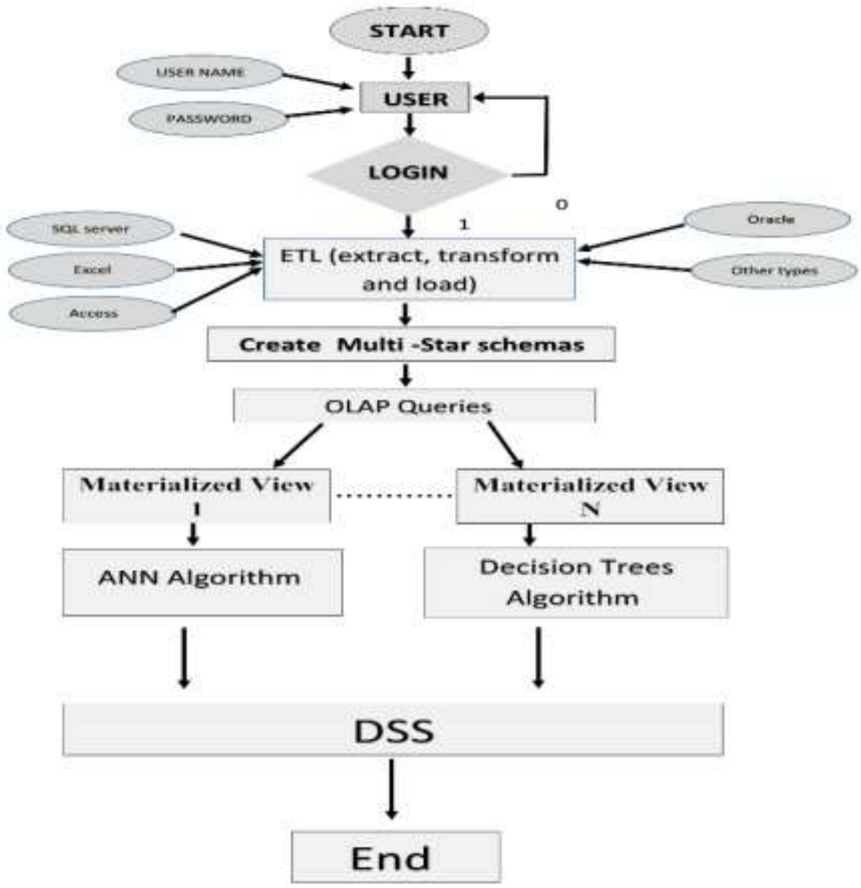

Figure 1. Flowchart of system design

5. On-Line Analytical Processing (OLAP)

OLAP is an approach using answering analytical queries, it is using in business intelligence, also using in a relational database, report writing, and data mining. In this section we have a tendency to suppose there square measure 3 complexes SQL OLAP queries with drill down operation, aggregation operations like (COUNT, SUM, MIN, MAX), join, selection, filtering operation like (using condition where) and cluster by operation doing to pick out information from tables within the university system as shown below:

1 - The primary report is: count the total range of scholars within the university and its classification to college, departments, and section.

2- The second report is: count the entire range of graduates students within the department, classification to gender (male or female) [10].

Algorithm (1) the proposed OLAP algorithm

Input: OLAP query, set of constraints

Output: report contains information according to query constraints

1. Start
2. User send OLAP query through application tools.
3. Query contains constraint deterministic.
4. Assigned parameters that needing for analyzing.
5. OLAP engine specified data to OLAP query.
6. Select the dimension from dimension hierarchies and measure data.
7. OLAP engine sent the result to user interface.
8. User receiving request (REPORT) from the interface application.
9. The report content attributes and hugs of columns.
10. End start.

6. Materialized view

Given an MVPP, we shall find a set of materialized views such that the total cost for query processing and view maintenance MVPP is the materialized operation for the query processing in the obtained combinations. It obtained the mixed area and its model for the development of the query. The query based model has the following functional attributes which elaborated the specific view and algorithm.

1. O_v represent the universal queries for usability of v, O_v =R∩D*{v}; where R is the set of root nodes and D*{v} is the set of ancestors of v as defined in Section 3.
2. $I_v$denotes the base relations which are used to produce v, I_v= L∩S*{v}; where L is the set of leaf nodes and S*{v} is the set of descendants of v as defined in Section 3.
3. The weight of the concerned node is w(v) calculated as w(v) = $\sum_{q\in O_V} f_q$(q)*C_a^q(v)-$\sum_{r\in I_V} f_u\,(r) * C_m^r(v)$.

The mathematical formula and expression part shows the v is the special view of the database and its storage of the database.[9]

Algorithm (2) materialized view

Input: Data from tables

Output: Save Summary of data

Begin

1. M :=Ø;
2. create list LV for all the nodes (with positive value of weights) based on the descending order of their weights;
3. pick up the first one v from LV ;
4. generate O_v , I_v , and S_v;
5. calculate $C_s = \sum_{q\in O_V}\{f_q(q)*(C_a^q(v)- \sum_{u\in S_V\cap M} C_a^q(u))\}- \sum_{r\in I_V}\{f_u(r)C_m^r(v)\}$;
6. if $C_s > 0$, then
   6.1. insert v into M ;
   6.2. remove v from LV ;
7. else remove v and all the nodes Listed after v from LV who are in The subtree rooted at v;
8. repeat step 3 until LV = Ø;
9. for each v ∈ M , if D(v) ⊂ M , then Remove v from M;
   End;

- LV node value is list among all based on decrying order of w (v).
- S_v = S*{v} node is responsible for the production of the value of v
- Suppose number of queries in the execution is k;
- Assign as M becomes the materialized view of the query.

Without losing generality, we assume there are altogether k queries. Let M be the set of materialized views. The algorithm 2 for determining M is based on the following idea: whenever a new node is considered to be materialized, we calculate the saving it brings in accessing all the queries involved, subtracting the cost for maintaining this node. If this value is positive, then this node will be materialized and added into M.
From step 5, the C_s having the value of v which is a materialized view. The second part is the view maintenance cost for v. $C_s > 0$ 0 shows that there is a cost gain if v is materialized. $\sum_{u\in S_V\cap M} C_a^q(u))$ is the replicated saving in case of some descendants of v are already chosen to be materialized. After applying transformation, C_s becomes:
$C_s=\sum_{q\in O_V} f_q(q)*(C_a^q(v)- \sum_{r\in I_V} f_u(r)C_m^r(v)- \sum_{q\in O_V} f_q(q) \sum_{u\in D_{(V)}\cap M} C_a^q(u))$
$=w(v)- \sum_{q\in O_V} f_q(q) * \sum_{u\in D_{(V)}\cap M} C_a^q(u))$
From the given above example of query, the v1 and v2 are calculated from above part of a mathematical expression. The special view of each query is calculated in step 7 from above algorithm. [٩]

7. Decision Tree Algorithm

The model- (or tree-) building side of decision tree classification algorithms area unit composed of two main tasks: tree induction and tree pruning. Tree induction is that the task of taking a group of pre-classified instances as input, deciding that attributes are best to separate on, cacophonous the dataset, and recuing on the ensuing split datasets till all coaching instances area unit categorized.

While building our tree, the goal is to separate on the attributes that produce the purest kid nodes attainable, which might keep to a minimum the quantity of splits that will have to be compelled to be created so as to classify all instances in our dataset. This purity is usually measured by one in every of variety of various attribute choice measures. [8]

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). decision tree algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has an entropy of one.

Entropy (S) = - p+ log(p+) – p-log(p)

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

Gain (S,A)=Entropy(s) - $\sum_{v\, \in value\ (A)} \frac{|S_v|}{|S|} . Entropy\ (S_v)$ [15]

Algorithm (3) Decision Tree Algorithm

Input: S, where S=set of classified instances
Output: Decision Tree

Require: S ≠Ø, num_attributes >0
1. Procedure BULLDTREE
2. Repeat
3. maxGain ← 0
4. splitA ← null
5. e ← Entropy (Attributes)
6. for all Attributes a in S do
7. gain ← informationGain (a, e)
8. if gain > maxGain then
9. maxGain ← gain
10. splitA ← a
11. end if
12. end for
13. Partition(S, splitA)
14. Until all partitions processed
15. End Procedure

8. Artificial Neural Network Algorithm[8]

The Neural network is the most efficient algorithm in computational technology. As we understand the algorithm in straight forwards which belong to following steps.

Algorithm (4) Neural Network Algorithm

Input:

- D, a data set consisting of the training tuples and their associated target values;
- l, the learning rate;
- network, a multilayer feed-forward network

Output: A trained neural network.

Initialize all weights and biases in *network*;
while terminating condition is not satisfied {
for each training tuple *X* in *D* {
// Propagate the inputs forward:
for each input layer unit j {
$O_j = I_j$; // output of an input unit is its actual input value
for each hidden or output layer unit *j* {
$I_j = \sum_i W_{ij} O_j + \theta_j$; //compute the net input of unit j with respect to The previous layer, *i*
$O_j = \frac{1}{1+e^{-I_j}}$; }// compute the output of each unit j
// Backpropagate the errors:
for each unit j in the output layer
$Err_j = O_j(1 - O_j)(T_i - O_j)$; // compute the error
for each unit j in the hidden layers, from the last to the first hidden layer
$Err_j = O_j(1 - O_j) \sum_k Err_j W_{jk}$; // compute the error with respect to The next higher layer, k
for each weight W_{ij} in network {
$\Delta W_{ij} = (l) Err_j O_i$; // weight increment
$W_{ij} = W_{ij} + \Delta W_{ij}$; } // weight update
for each bias θ_j; in network{
$\Delta\theta_j = (l) Err_j$; // bias increment.
$\theta_j = \theta_j + \Delta\theta_j$; }// bias update
}}

Initial values of the weight in the outcome network are numerical values. The weights within the network square degree initialized to little numeric numbers (e.G., starting ranges −1.0 to at least one.0, or −zero.Five to zero.Five). Every unit includes a bias related to it, as explained later. The biases square degree equally initialized to little random numbers.

Each education tuple, X, is processed by way of the following steps.

Propagate the inputs forward: First, the training tuple is fed to the network's input layer. The inputs pass through the input units, unchanged. That is, for an input unit, j.

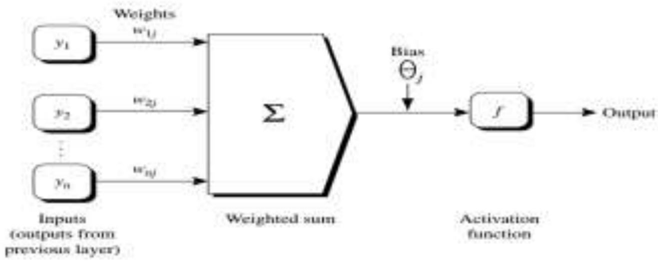

Figure (2) Non-linear model of a neuron

Hidden or output layer unit j: The inputs to unit j are outputs from the previous layer. These are multiplied by their corresponding weights to form a weighted sum, which is added to the bias associated with unit j. A nonlinear activation function is applied to the net input. (For ease of explanation, the inputs to unit j are labeled y1, y2,…., yn. If unit j were in the first hidden layer, then these inputs would correspond to the input tuple .x1, x2,…, xn.)

its output, Oj, is equal to its input value, I_j. Next, the net input and output of each unit in the hidden and output layers are computed. The net input to a unit in the hidden or output layers is computed as a linear combination of its inputs. To help illustrate this point, a hidden layer or output layer unit is shown in Figure 2. Each such unit has a number of inputs to it that are, in fact, the outputs of the units connected to it in the previous layer. Each connection has a weight. To compute the net input to the unit, each input connected to the unit is multiplied by its corresponding weight, and this is summed. Given a unit, j in a hidden or output layer, the net input, I_j, to unit j is

$$I_j = \sum_i W_{ij}\, O_i + \theta_j$$

Where W_{ij} is the weight of the connection from unit i in the previous layer to unit j; O_i is the output of unit i from the previous layer; and θ_j is the bias of the unit. The bias acts as a threshold in that it serves to vary the activity of the unit.

Each unit in the hidden and output layers takes its net input and then applies an activation function to it, as illustrated in Figure 2. The function symbolizes the activation of the neuron represented by the unit. The logistics, or sigmoid, a function is used. Given the net input I_j to unit j, then O_j, the output of unit j, is computed as

$$O_i = \frac{1}{1+ e^{-I_j}}$$

This function is also referred to as a squashing function because it maps a large input domain onto the smaller range of 0 to 1. The logistic function is nonlinear and differentiable, allowing the backpropagation algorithm to model classification problems that are linearly inseparable.

We compute the output values, O_j, for each hidden layer, up to and including the output layer, which gives the network's prediction. In practice, it is a good idea to cache (i.e., save) the intermediate output values at each unit as they are required again later when backpropagating the error. This trick can substantially reduce the amount of computation required.

Backpropagate the error: The error is propagated backward by updating the weights and biases to reflect the error of the network's prediction. For a unit j in the output layer, the error Err_j, is computed by

$$Err_j = O_i(1 - O_i)(T_i - O_i)$$

where O_j is the actual output of unit j, and T_j is the known target value of the given training tuple. Note that O_j.1 - O_j/ is the derivative of the logistic function.

To compute the error of a hidden layer unit j, the weighted sum of the errors of the units connected to unit j in the next layer are considered. The error of a hidden layer unit j is

$$Err_j = O_i(1 - O_i)\sum_k Err_k W_{jk}$$

where W_{ij} is the weight of the connection from unit j to a unit k in the next higher layer, and Err_K is the error of unit k.

The weights and biases are updated to reflect the propagated errors. Weights are updated by the following equations, where ΔW_{ij} is the change in weight W_{ij}:

$$\Delta W_{ij} = (1)\, Err_j\, O_i$$

$$W_{ij} = W_{ij} + \Delta W_{ij}$$

"What is l in Eq." The variable l is the learning rate, a constant typically having a value between 0.0 and 1.0. Backpropagation learns using a gradient descent method to search for a set of weights that fits the training data so as to minimize the mean squared distance between the network's class prediction and the known target value of the tuples.1 The learning rate helps avoid getting stuck at a local minimum in decision space (i.e., where the weights appear to converge but are not the optimum solution) and encourages finding the global minimum. If the learning rate is too small, then learning will occur at a very slow pace. If the learning rate is too large, then oscillation between inadequate solutions may occur. A rule of thumb is to set the learning rate to 1/t, where t is the number of iterations through the training set so far.

Biases are updated by the following equations, where $\Delta\theta_j$ is the change in bias θ_j:

$$\Delta\theta_j = (1) Err_j$$

$$\theta_j = \theta_j + \Delta\theta_j$$

Note that here we are updating the weights and biases after the presentation of each tuple. This is referred to as case updating. Alternatively, the weight and bias increments could be accumulated in variables, so that the weights and biases are updated after all the tuples in the training set have been presented. This latter strategy is called epoch updating, where one iteration through the training set is an epoch. In theory, the mathematical derivation of backpropagation employs epoch updating, yet in practice, case updating is more common because it tends to yield more accurate results.

Terminating condition: Training stops when

All ΔW_{ij} in the previous epoch are so small as to be below some specified threshold, or

The percentage of tuples misclassified in the previous epoch is below some threshold, or

A prespecified number of epochs has expired

In practice, several hundreds of thousands of epochs may be required before the weights will converge.

"How efficient is backpropagation?" The computational efficiency depends on the time spent training the network. Given |D| tuples and w weights, each epoch requires O(|D| × w) time. However, in the worst-case scenario, the number of epochs can be exponential in n, the number of inputs. In practice, the time required for the networks to converge is highly variable. A number of techniques exist that help speed up the training time. For example, a technique known as simulated annealing can be used, which also ensures convergence to a global optimum. [8]

9. Result Discussion

The results of applying two algorithms for access to decision support system (Artificial neural network and decision tree algorithm).As shown in the table(1) time method algorithms has been applied on completely different sizes of data warehouse that are (15251, 52364, 104728, 2880020, 7488052) records. it had been noticed that once the data warehouse size(the

Database size for processing	Processing time	Rate of inconsistent records that processed	Rate of processing (consistency)
١٥٢٥١	0:00:05	١٤٢٦٤/١٥٢٥١	93.52
52364	0:00:11	42962/52364	82.04
104728	0:00:14	82595/104728	78.86
2880020	0:01:54	2024612/2880020	70.29
7488052	0:04:52	5042765/7488052	67.34

number of records) will increase, the overwhelming time of the process is accrued however still deficient.

9.1 Time processing

By applying the algorithms on data warehouse size 15251 records, the processing time that it takes is just (5 seconds) for access to decision support system from materialized views size (14264), on data warehouse size 52364 records , the processing time that it takes is just (11 seconds) for access to decision support system from materialized views size (42962), on data warehouse size 104728 records , the processing time that it takes is just (14 seconds) for access to decision support system from materialized views size (82595), on data warehouse size 2880020 records , the processing time that it takes is just (one minutes and 54 seconds) for access to decision support system from materialized views size (2024612), on data warehouse size 7488052 records, the processing time that it takes is just (four minutes and 52 seconds) for access to decision support system from materialized views size (5042765).

Table 1 Time process of Executing algorithms

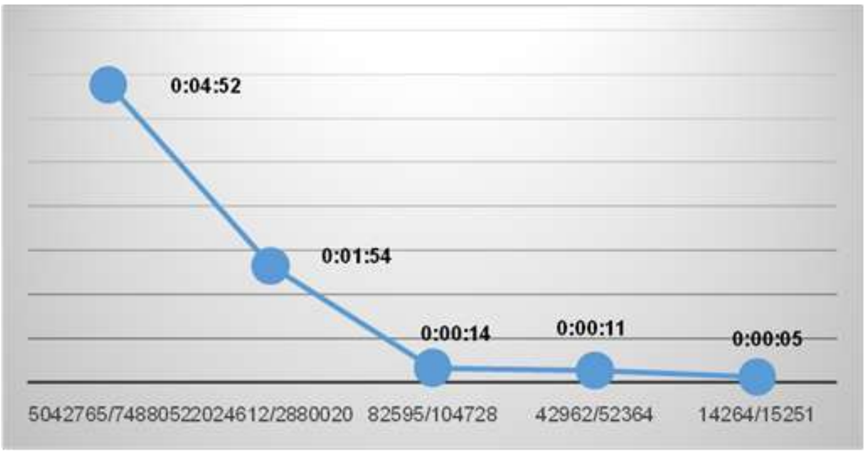

Figure (3) Time required to implement algorithms

9.2 Result of algorithms for decision support system

Figure 4 artificial neural network shows that the highest percentages of graduates were male students. In the Computer Science Department, the percentage of graduates for all the years was 39.27% for females and 60.73% for males. The English Department was 36.4% for females and 63.4% for males. The law Department percentage of males is 3.79%, the male percentage is 93.21, the Arabic language Department is female, 21.93%, the male is 78.07%, the computer technology-engineering department is the female ratio of 17.55%, the male ratio is 82.45%, the financial and banking sciences department is 14.15% and the male is 85.85%.

The rates of graduates between 82.873 - 100 females were significantly lower than the male rate of 17.31 for females and 82.67 for males. And the rates between the highest - 50 - 64.934 in these rates were for males 91% and for females 8.98, while the age of students between 34-39 was the proportion of females and 12.50 male remaining rate of the students between 73.904 - 82.873 female ratio 87.36 and the remaining percentages were for men.

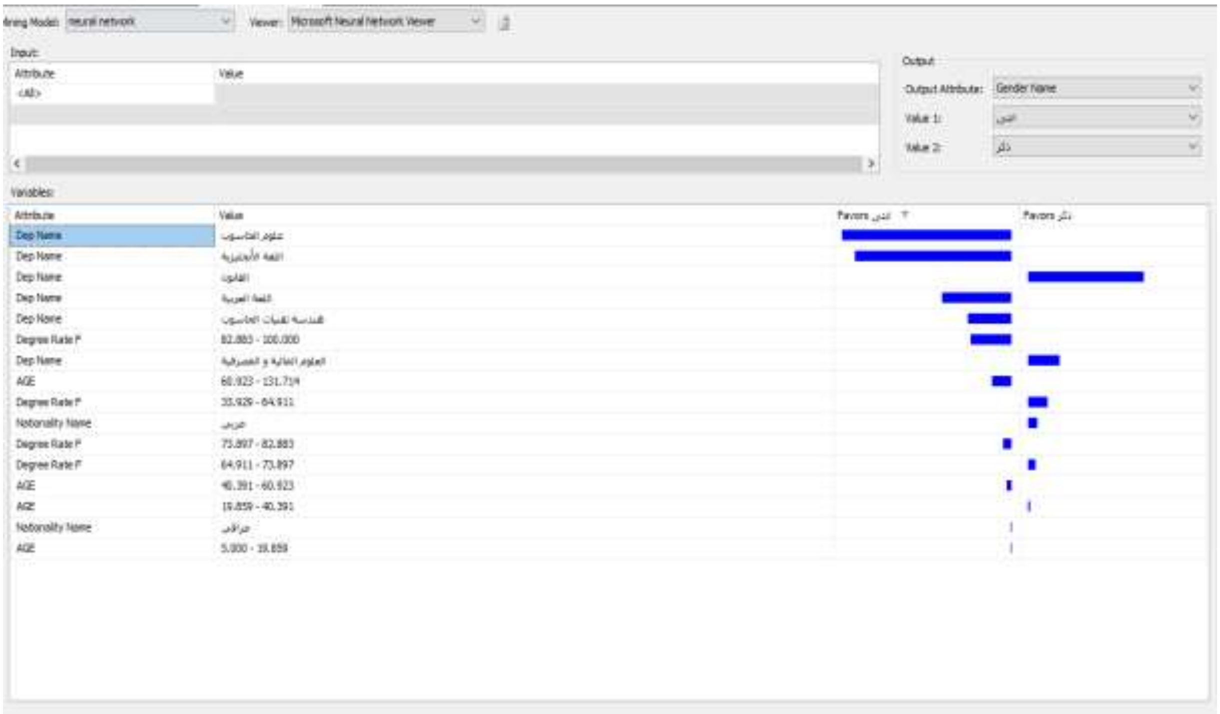

Figure (4) Neural network

The Figure 5 decision tree show :the main node for all collage graduates = 7483 , 6318 from male and 1165 from female ,divide to six department .the first department: banking sciences department 8.92 of graduates student from female and 91.08 from male .second department: English language department 42.63 % from female and 57.37 from male .the third department computer Science 21.54% from female and 78.46 from male .the fourth department: Arabic language department 25.40 from female and 74.60 from male ,the fifth Department: law Department 3.79% female and 96.21 from male and the last department: computer technology-engineering department 13.73% female and 86.27% male .

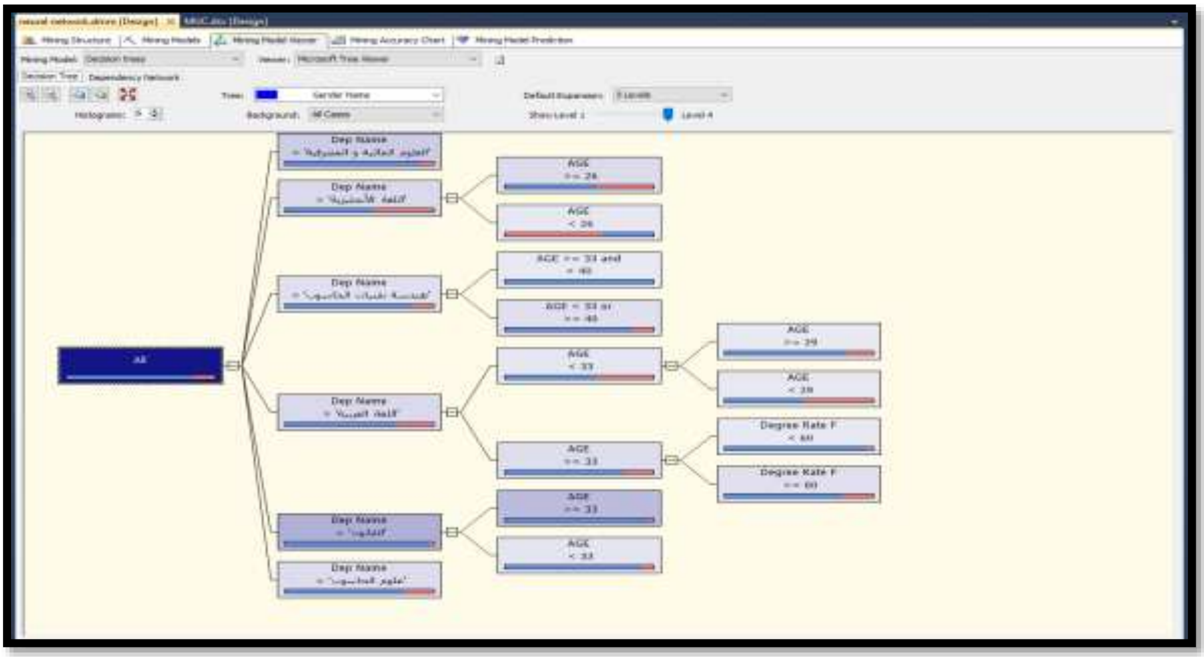

Figure (5) Decision tree

10. Conclusions

After the implementation of the proposed Design Dynamic Data Warehouse system and through the execution of the proposed algorithms of creating data warehouse which is building facts and dimension tables that can help to improve the proposed system, and clarify the Multi-Star schema relationship between fact and dimension tables, and use hybrid algorithm artificial neural network and decision tree algorithm for access to good decision support system:

1. Database used in different fields of works, due to the multiple and different using of databases, it needs to enhance operations to change its structure in terms of indexes or materialized views to speed up and increase the efficiency of paging operations.
2. Systems that have a large number of items and all these items that contain a large number of ranges; are a good option to test the operations that's related to database performance.
3. The most common way to improve database performance is to change the materialized views, but the more effective way is to change the way of writing the query.
4. An artificial intelligence algorithm on one hand is needed and on the other hand, is a simple way of writing the query and deliver them with a meaningful manner.
5. Artificial neural network and decision tree is an artificial intelligence algorithm that is highly efficient to find a stable solution from among multiple options for solutions.
6. The efficiency of a proposed artificial neural network and Decision tree algorithm to process large data warehouse size with (7488052 records) at a very little time that is (4:52 seconds).

4. References

[1] Payal Pahwa, Rashmi Chhabra, "An Object Oriented Data Warehouse Design ", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4, Issue-ICCIN-2K14, March 2014.

[2] Inmon WH. "Building the data warehouse (second edition)". John Wiley and Sons; 1996

[3] Dimitri Theodoratos Timos Sellis," Dynamic Data Warehouse Design", Research supported by the European Commission under the ESPRIT Program LTR project "DWQ: Foundations of Data Warehouse Quality, 2000.

[4]J.C. Mostert, M.M.M. Snyman," Knowledge management framework for the development of an effective knowledge management strategy ", Vol.9 (2) June 2007.

[5] M.Hamad, and Y.Turky," A Dynamic Warehouse Design Based on Simulated Annealing Algorithm", Vol 6, No 1 (2016)

[6] Hamid R. Nemati, David M. Steiger, Lakshmi S. Iyer, and Richard T. Herschel, "Knowledge Warehouse: An Architectural Integration of Knowledge Management, Decision Support, Data Mining and Data Warehousing", 2009, University of North Carolina at Greensboro.

[7] N. Kasabov, D. Deng , L. Erzegovezi , M. Fedrizzi , A. Beber ," Hybrid Intelligent Decision Support Systems for Risk Analysisand Prediction of Evolving Economic Clusters in Europe ", Pages 347-372 (2000)

[8] Jiawei Han, Micheline Kamber, Jian Pe," Data Mining Concepts and Technique", Third Edition(2000).

[9] Jian Yang, Kamalakar Karlapalem, Qing Li "Algorithms for Materialized View Design in Data Warehousing Environment ", Pages 136-145 , August 25 - 29, 1997 .

[10] Thesis: Yousra Atalla Turky, " A Dynamic Warehouse Design Using Simulation Modeling Approach", university of anbar ,2015

[11] Rania Yangui, Ahlem Nabli, Faiez Gargouri," DW4SN: A Tool for Dynamic Data Warehouse Building from Social Network ", Vol. 134, pp. 191-205, 2017.

[12]Wan Hussain Wan Ishak, Ku Ruhana Ku-Mahamud , and Norita Md Norwawi , their research is named " Intelligent Decision Support Model Based on Neural Network to Support Reservoir Water Release Decision ", ICSECS 2011, Part I, CCIS 179, pp. 365–379, 2011.

[13] GarimaSahu, Rakesh Kumar Khare," Decision Tree Classification based Decision Support System for Derma Disease", International Journal of Computer Applications (0975 – 8887),Volume 94 – No.17, May 2014.

[14] Han, J., Kamber, M., and Pei, J. (2011).Data mining: Concepts and Techniques, 3rd edition, Morgan Kaufmann, San Francisco, CA. USA.

[15] "Decision Tree", https://www.saedsayad.com/decision_tree.htm