World of Computer Science and Information Technology Journal (WCSIT)
ISSN: 2221-0741
Vol. 1, No. 7, 311-316, 2011

An Improved Algorithm for Mining Association Rules in Large Databases

Farah Hanna AL-Zawaidah
Department of Computer Information System
Irbid National University
Irbid, Jordan

Yosef Hasan Jbara
Computer Technology Department
Yanbu College of Technology
Yanbu, Saudi Arabia

Marwan AL-Abed Abu-Zanona
Department of Computer Information System
Jerash University
Amman, Jordan

Abstract— Mining association rules in large databases is a core topic of data mining. Discovering these associations is beneficial to the correct and appropriate decision made by decision makers. Discovering frequent itemsets is the key process in association rule mining. One of the challenges in developing association rules mining algorithms is the extremely large number of rules generated which makes the algorithms inefficient and makes it difficult for the end users to comprehend the generated rules. This is because most traditional association rule mining approaches adopt an iterative technique to discover association rule, which requires very large calculations and a complicated transaction process. Furthermore, the existing mining algorithms cannot perform efficiently due to high and repeated disk access overhead. Because of this, in this paper we present a novel association rule mining approach that can efficiently discover the association rules in large databases. The proposed approach is derived from the conventional Apriori approach with features added to improve data mining performance. We have performed extensive experiments and compared the performance of our algorithm with existing algorithms found in the literature. Experimental results show that our approach outperforms other approaches and show that our approach can quickly discover frequent itemsets and effectively mine potential association rules.

Keywords-mining; association rules; frequent patterns; apriori.

I. INTRODUCTION

In the recent years it has seen a dramatic increase in the amount of information or data being stored in database. The exponentially growth in size of on hand databases, mining for latent knowledge become essential to support decision making. Data mining is the key step in the knowledge discovery process. The main tasks of Data mining are generally divided in two categories: Predictive and Descriptive. The objective of the predictive tasks is to predict the value of a particular attribute based on the values of other attributes, while for the descriptive ones, is to extract previously unknown and useful information such as patterns, associations, changes, anomalies and significant structures, from large databases. There are several techniques satisfying these objectives of data mining. Some of these can be classified into the following categories: clustering, classification, association rule mining, sequential pattern discovery and analysis. The development of data mining systems has received a great deal of attention in recent years. It plays a key enabling role for competitive businesses in a wide variety of business environments. It has been extensively applied to a wide variety of applications like sales analysis, healthcare, Ecommerce, manufacturing, etc. A number of studies have been made on efficient data mining methods and the relevant applications. In this study we considered Association Rule Mining for knowledge discovery and generate the rules by applying our developed approach on real and synthetic databases.

Mining Associations is one of the techniques involved in the process mentioned above and among the data mining problems it might be the most studied ones. Discovering association rules is at the heart of data mining. Mining for association rules between items in large database of sales transactions has been recognized as an important area of database research [1]. These rules can be effectively used to uncover unknown relationships, producing results that can provide a basis for forecasting and decision making. The original problem addressed by association rule mining was to find a correlation among sales of different products from the analysis of a large set of supermarket data. Today, research work on association rules is motivated by an extensive range of application areas, such as banking, manufacturing, health care, and telecommunications. It is also used for building statistical thesaurus from the text databases [2], finding web access patterns from web log files [3], and also discovering associated images from huge sized image databases [4].

A number of association rule mining algorithms have been developed in the last few years [5, 6, 7, 8, 9, 10], which can be classified into two categories: (a) candidate generation/test approach such as Apriori [6] (b) pattern growth approach [9, 10]. A milestone in the first category studies is the development of an Apriori based, level wise mining method for associations, which has sparked the development of various kinds of Apriori like association mining algorithms. Among these, the Apriori algorithm has been very influential. Since its inception, many scholars have improved and optimized the Apriori algorithm and have presented new Apriori like algorithms [11, 12, 13, 14, 15]. The Apriori like algorithms adopt an iterative method to discover frequent itemsets.

The existing mining algorithms have some drawbacks: Firstly, the existing mining algorithms are mostly designed in forms of several passes so that the whole database needs to be read from disks several times for each user's query under the constraint that the whole database is too large to be stored in memory. This is very inefficient in considering the big overhead of reading the large database even though only partial items are interested in fact. As a result, they cannot perform efficiently in terms of responding the user's query quickly. Secondly, in many cases, the algorithms generate an extremely large number of association rules, often in thousands or even millions. Further, the association rules are sometimes very large. It is nearly impossible for the end users to comprehend or validate such large number of complex association rules, thereby limiting the usefulness of the data mining results. Thirdly, no guiding information is provided for users to choose suitable settings for the constraints such as support and confidence such that an appropriate number of association rules are discovered. Consequently, the users have to use a try and -error approach to get suitable number of rules. This is very time consuming and inefficient. Therefore, one of the main challenges in mining association rules is developing fast and efficient algorithms that can handle large volumes of data.

In this paper we attack the association rule mining by an apriori based approach specifically designed for the optimization in very large transactional databases. The developed mining approach called Feature Based Association Rule Mining Algorithm (FARMA).

The rest of the paper is organized as follows. We introduce the theoretical properties and formal definition of association rule in section 2. Section 3, presents some related work. . In Section 4, the proposed method is described in details. Section 5 presents comparisons and experiments results of our proposed approach with various Apriori algorithms and other algorithms found in literature. Finally, a conclusion and the future work are given in Section I.

II. Association Rules Mining

Association is the discovery of association relationships or correlations among a set of items. This problem was introduced in [5]. Let I = {i1, i2, …, im} be a set of binary attributes, called items. Let D a set of transactions and each transaction T is a set of items such that $T \subseteq I$. Let X be a set of items. A transaction T is said to contain X if and only if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. Furthermore, the rule $X \Rightarrow Y$ is said to hold in the transaction set D with confidence c if there are c% of the transaction set D containing X also containing Y. The rule $X \Rightarrow Y$ is said to have support s in the transaction set D if there are s% of transactions in D containing $X \cup Y$. An example of an association rule is: "35% of transactions that contain bread also contain milk; 5% of all transactions contain both items". Here, 35% is called the confidence of the rule, and 5% the support of the rule [5, 16]. The selection of association rule is based on support and confidence. The confidence factor indicates the strength of the implication rules, i.e. the confidence for an association rule is the ratio of the number of transactions that contain X U Y to the number of transactions that contain X; whereas the support factor indicates the frequencies of the occurring patterns in the rule. i.e., the support for an association rule is the percentage of transactions in the database that contain X U Y. Given the database D, the problem of mining association rules involves the generation of all association rules among all items in the given database D that have support and confidence greater than or equal to the user specified minimum support and minimum confidence. Typically large confidence values and a smaller support are used. Rules that satisfy both minimum support and minimum confidence are called strong rules. Since the database is large and users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful.

The discovery of association rules for a given dataset D is typically done in two steps [5, 16]: discovery of frequent itemsets and the generation of association rules. The first step is to find each set of items, called as itemsets, such that the co-occurrence rate of these items is above the minimum support, and these itemsets are called as large itemsets or frequent itemsets. In other words, find all sets of items (or itemsets) that have transaction support above minimum support. Finding large itemsets is generally very easy but very costly. The naive approach would be to count all itemsets that appear in any transaction. Suppose one of the large itemsets is Lk, Lk = {I1, I2, …, Ik}, association rules with this itemsets are generated in the following way: the first rule is {I1, I2, … , Ik1} ⇒ {Ik}, by checking the confidence this rule can be determined as interesting or not. Then other rule are generated by deleting the last items in the antecedent and inserting it to the consequent, further the confidences of the new rules are checked to determine the interestingness of them. Those processes iterated until the antecedent becomes empty. The size of an itemset represents the number of items in that set. If the size of an itemset is equal to k, then this itemset is called as the k itemset. The second step is to find association rules from the frequent itemsets that are generated in the first step. The second step is rather straightforward. Once all the large itemsets are found generating association rules is straightforward. The first step dominates the processing time and for that reason, it has been one of the most popular research fields in data mining. So, we explicitly focus this paper on the first step.

Generally, an association rules mining algorithm contains the following steps [17]:

- The set of candidate k itemsets is generated by 1-extensions of the large (k 1)itemsets generated in the previous iteration.
- Supports for the candidate k itemsts are generated by a pass over the database.
- Itemsets that do not have the minimum support are discarded and the remaining itemsets are called large k itemsets.

This process is repeated until no more large itemsets are found.

III. PREVIOUS WORK

The problem of discovering association rules was first introduced in [5] and an algorithm called AIS was proposed for mining association rules. For last fifteen years many algorithms for rule mining have been proposed. Most of them follow the representative approach by Agrawal et al. [16], namely Apriori algorithm. Various researches were done to improve the performance and scalability of Apriori included using parallel computing. There were also studies to improve the speed of finding large itemsets with hash table, map, and tree data structures. Here we review some of the related work that forms a basis for our algorithm.

A. AIS Algorithm

The AIS algorithm was the first algorithm proposed for mining association rules [5]. The algorithm consists of two phases. The first phase constitutes the generation of the frequent itemsets. The algorithm uses candidate generation to detect the frequent itemsets. This is followed by the generation of the confident and frequent association rules in the second phase. The main drawback of the AIS algorithm is that it makes multiple passes over the database. Furthermore, it generates and counts too many candidate itemsets that turn out to be small, which requires more space and wastes much effort that turned out to be useless.

B. Apriori Algorithms

The Apriori algorithm from [16] is based on the Apriori principle, which says that the itemset X' containing itemset X is never large if itemset X is not large. Based on this principle, the Apriori algorithm generates a set of candidate large itemsets whose lengths are (k+1) from the large k itemsets (for k≥1) and eliminates those candidates, which contain not large subset. Then, for the rest candidates, only those with support over minsup threshold are taken to be large (k+1)itemsets. The Apriori generate itemsets by using only the large itemsets found in the previous pass, without considering the transactions.

The AprioriTid [16] algorithm is a variation of the Apriori algorithm. The AprioriTid algorithm also determines the candidate itemsets before the pass begins. The main difference from the Apriori algorithm is that the AprioriTid algorithm does not use the database for counting support after the first pass. Instead, the large k itemset in the transaction with identifier TID is used for counting. The downside of using this scheme for counting support is that the large itemsets that would have been generated at each pass may be huge. Another algorithm, called Apriori Hybrid, is introduced in [16]. The basic idea of the Apriori Hybird algorithm is to run the Apriori algorithm initially, and then switch to the AprioriTid algorithm when the generated database, i.e. large k itemset in the transaction with identifier TID, would fit in the memory.

C. DHCP Algorithm

The DHP (Direct Hashing and Pruning) algorithm [12] is an effective hash based algorithm for the candidate set generation. It reduced the size of candidate set by filtering any k itemset out of the hash table if the hash entry does not have minimum support. The hash table structure contains the information regarding the support of each itemset. The DHP algorithm consists of three steps. The first step is to get a set of large 1-itemsets and constructs a hash table for 2itemsets. The second step generates the set of candidate itemsets Ck. The third step is the same as the second step except it does not use the hash table in determining whether to include a particular itemset into the candidate itemsets. Furthermore, it should be used for later iterations when the number of hash buckets with a support count greater than or equal to the minimum transaction support required is less than a predefined threshold.

D. Partition Algorithm

The Partition algorithm [18] logically partitions the database D into n partitions, and requires just two database scans to mine large itemsets. The algorithm consists of two phases. In the first phase, the algorithm subdivides the database into n no overlapping partitions which can fit into main memory. The algorithm iterates n times, and during each iteration only one partition is considered. In the second phase, the algorithm counts actual support of each global candidate itemsets and generates the global large itemsets.

E. AIS Algorithm

The frequent pattern growth (FP growth) algorithm [19] improves Apriori by using a novel data structure which is the frequent pattern tree, or FP tree. It stores information about the frequent patterns. The algorithm adopts a divide and conquer strategy and a frequent pattern tree to mine the frequent patterns with only two passes over the database and without candidate generation which is a big improvement over Apriori.

IV. PROPOSED APPROACH

The developed approach adopts the philosophy of Apriori approach with some modifications in order to reduce the time execution of the algorithm. First, the idea of generating the feature of items is used and; second, the weight for each candidate itemset is calculated to be used during processing.

The feature array data structure is built by storing the decimal equivalent of the location of the item in the transaction. In other words transforming the transaction database into the feature matrix. Transforming here means reorganizing and transforming a large database into manageable structure to fulfill two objectives: (a) reducing the number of I/O accesses in data mining, and (b) speeding up the mining process. There is one mandatory requirements for the transforming technique, that the transaction database should be read only once within the whole life cycle of data mining. By storing the appearing feature of each interested item as a compressed vector separately, the size of the database to be accessed can be reduced greatly. To calculate the weight for each candidate itemset Ck, the developed approach scans the array data structure and the items contained in Ck are accessed and the weight is obtained by summing the decimal equivalent of each item in the transaction. Similar process is done for calculating the support value for each item. To calculate the support value for each candidate itemset Ck, the developed approach scans the array data structure and the items contained in Ck are accessed for and the value of support is obtained by counting the number of decimal equivalent appeared in the transaction. If a certain number of generations have not passed then repeat the process from the beginning otherwise generate the large itemsets by doing the union of all Lk. Once the large itemsets and their supports are determined, the rules can be discovered in a straight forward manner as follows: if I is a large itemset, then for every subset a of I, the ratio support (l) / support (a) is computed. If the ratio is at least equal to the user specified minimum confidence, then the rule a ⇒ (1a) is output. Multiple iterations of the discovery algorithm are executed until at least N itemsets are discovered with the user specified minimum confidence, or until the user specified minimum support level is reached. The algorithm uses Leverage measure introduced by Piatetsky [20] to filter the found item sets and to determine the interestingness of the rule.

$$\text{Leverage}(X \longrightarrow Y)=P(X \text{ and } Y)-(P(X)P(Y))$$

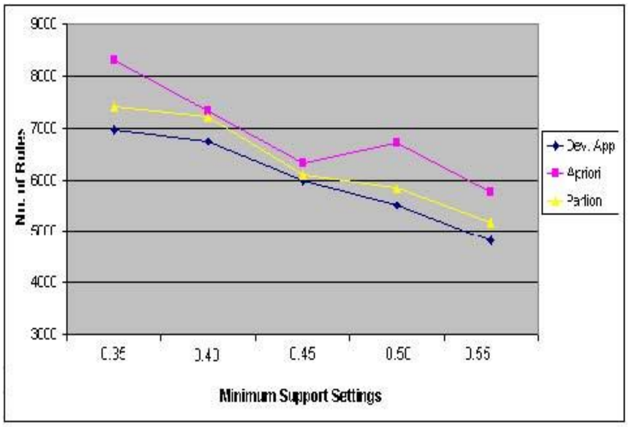
Leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y where statistically dependent. Using minimum leverage thresholds at the same time incorporates an implicit frequency constraint. e.g., for setting a min. leverage thresholds to 0.01% (corresponds to 10 occurrence in a data set with 100,000 transactions) one first can use an algorithm to find all itemsets with minimum support of 0.01% and then filter the found item sets using the leverage constraint. By using Leverage measure we reduce the generation of candidate's itemsets and thus we reduce the memory requirements to store a huge number of useless candidates. This is one of the main contributions of this paper.

V. EXPERIMENTAL RESULTS

To evaluate the efficiency of the proposed method we have extensively studied our algorithm's performance by comparing it with the Apriori algorithm as well as partition algorithm and consider the superiority of it. All algorithms are implemented using Microsoft Visual Basic for Applications (VBA) and run on a 3.2 GHz Pentium 4 PC with 2 GB of RAM and 250GB Hard Disk running the XP operating system. The test database is the transaction database provided with Microsoft SQL Server 2000. Three data sets of 1200,000, 400,000, and 750,000 transaction records of experimental data are randomly sampled from the FoodMart transaction database. The test database contain 420, 557, and 682 items respectively; with the longest transaction record contains 18, 26, 38 items respectively and the lowest transaction record contains 5, 7, 11 items respectively. We studied the effect of different values of minimum support (Minsup) which are set at 0.35%, 0.40%, 0.45%, 0.50%, 0.55% and different values of minimum confidence (Minconf) which are set at 0.40%, 0.47%, 0.55%, 0.60%, 0.65% on the processing time for the algorithms. We have observed considerable reduction in the number of association rules generated by our algorithm, as compared to other competing algorithm. Table 1 presents our experimental results. This reduction is also dependent on the support and confident values used. In Figure 1 we compare different possible settings of minimum support (Minsup) by plotting the number of rules generated versus the possible settings of minimum support over dataset 2. The development of the number of rules generated in Figure1 show that the smaller the value of minimum support, the more the number of rules generated.

TABLE I. EXPERIMENTAL RESULTS

	Data set 1	***Data set 2***	***Data set 3***
no. of transactions	120,000	400,000	750,000
No. of items	420	557	682
Max items/ transaction	18	26	38
Min items/ transaction	5	7	11
Support %	0.35%, 0.40 %, 0.45%, 0.50%, 0.55%	0.35%, 0.40 %, 0.45%, 0.50%, 0.55%	0.35%, 0.40 %, 0.45%, 0.50%, 0.55%
Confidence %	0.40%, 0.47 %, 0.55 % 0.60%, 0.65%	0.40%, 0.47 %, 0.55 % 0.60%, 0.65%	0.40%, 0.47 %, 0.55 % 0.60%, 0.65%
Avg # rules / FARMA	6262	6823	8118
Avg # rules / Apriori	7005	7597	9011
Avg # rules /Partition	6618	7107	8391



Test Result of Mining Setting

Yet, with increasing value of minimum support to a reasonable value, the better the solution quality obtained (smaller number of rules generated). The difference is more notable in the figure. In general, a value for Minsup between 0.4 and 0.6 permits the algorithm to be flexible enough over all the datasets and seems to give a reasonable compromise between the solutions qualities obtained.

CONCLUSIONS

The aim of this paper is to improve the performance of the conventional Apriori algorithm that mines association rules by presenting fast and scalable algorithm for discovering association rules in large databases. The approach to attain the desired improvement is to create a more efficient new algorithm out of the conventional one by adding new features to the Apriori approach. The proposed mining algorithm can efficiently discover the association rules between the data items in large databases. In particular, at most one scan of the whole database is needed during the run of the algorithm. Hence, the high repeated disk overhead incurred in other mining algorithms can be reduced significantly. We compared our algorithm to the previously proposed algorithms found in literature. The findings from different experiments have confirmed that our proposed approach is the most efficient among the others. It can speed up the data mining process significantly as demonstrated in the performance comparison. Furthermore, gives long maximal large itemsets, which are better suited to the requirements of practical applications. We demonstrated the effectiveness of our algorithm using real and synthetic datasets. We developed a visualization module to provide users the useful information regarding the database to be mined and to help the user manage and understand the association rules. Future work includes: 1) Applying the proposed algorithm to more extensive empirical evaluation; 2) applying our developed approach to real data like retail sales transaction and medical transactions to confirm the experimental results in the real life domain; 3) Mining multidimensional association rules from relational databases and data warehouses (these rules involve more than one dimension or predicate, e.g. rules relating what a customer shopper buy as well as shopper's occupation); 4) Mining multilevel association rules from transaction databases (these rules involve items at different levels of abstraction).

ACKNOWLEDGMENT

This work was made possible by a grant from Jarash National University.

REFERENCES

[1] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, 2000.

[2] J. D. Holt and S. M. Chung, "Efficient Mining of Association Rules in Text Databases" CIKM'99, Kansas City, USA, pp. 234242,Nov. 1999.

[3] B. Mobasher, N. Jain, E.H. Han, and J. Srivastava, "Web Mining: Pattern Discovery from World Wide Web Transactions" Department of Computer Science, University of Minnesota, Technical Report TR96-050, (March, 1996).

[4] C. Ordonez, and E. Omiecinski, "Discovering Association Rules Based on Image Content" IEEE Advances in Digital Libraries (ADL'99), 1999.

[5] R. Agrawal, T. Imielinski, and A. Swami. "Mining association rules between sets of items in large databases". In Proceedings of the ACM SIGMOD International Conference on Management of Data (ACM SIGMOD '93), pages 207216, Washington, USA, May 1993.

[6] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. "Fast discovery of association rules. In Advances in Knowledge Discovery and Data Mining", pages 307328. AAAI Press, 1996.

[7] R. Bayardo and R. Agrawal. "Mining the most interesting rules". In Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD '99), pages 145154, San Diego, California, USA, August 1999.

[8] J. Hipp, U. Güntzer, and U. Grimmer. "Integrating association rule mining algorithms with relational database systems". In Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS 2001), pages 130137, Setúbal, Portugal, July 710 2001.

[9] R. Ng, L. S. Lakshmanan, J. Han, and T. Mah. "Exploratory mining via constrained frequent set queries". In Proceedings of the 1999 ACM-SIGMOD International Conference on Management of Data (SIGMOD '99), pages 556558, Philadelphia, PA, USA, June 1999.

[10] Y. Guizhen: "The complexity of mining maximal frequent itemsets and maximal frequent patterns", Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining , pages:343353,August 2004, Seattle, WA, USA.

[11] L. Klemetinen, H. Mannila, P. Ronkainen, et al. (1994) "Finding interesting rules from large sets of discovered association rules". Third International Conference on Information and Knowledge Management pp. 401407.Gaithersburg, USA.

[12] J. S. Park, M.S. Chen, and P.S. Yu. "An Effective HashBased Algorithm for Mining Association Rules". Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, CA, USA, 1995, 175186.

[13] H. Toivonen, "Sampling large databases for association rules". 22nd International Conference on Very Large Data Bases pp. 134–145. 1996.

[14] P. Kotásek and J. Zendulka, "Comparison of Three Mining Algorithms for Association Rules". Proc. of 34th Spring Int. Conf. on Modelling and Simulation of Systems (MOSIS'2000), Workshop Proceedings Information Systems Modelling (ISM'2000), pp. 8590. Rožnov pod Radhoštěm, CZ, MARQ. 2000.

[15] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns Candidate generation". In Proc. 2000 ACMSIGMOD Int. Management of Data (SIGMOD'00), Dallas, TX. 2000.

[16] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Prof. 20th Int'l Conf. Very Large Data Bases, pp. 478499, 1994.

[17] K. Sotiris, and D. Kanellopoulos, "Association Rules Mining: A Recent Overview. GESTS International Transactions on Computer Science and Engineering", Vol.32 (1), 2006, pp. 7182.

[18] A. Savasere, E. Omiecinski, and S. Navathe. "An Efficient Algorithm for Mining Association Rules in Large Databases". Proceedings of 21th International Conference on Very Large Data Bases (VLDB'95),

September 1115, 1995, Zurich, Switzerland, Morgan Kaufmann, 1995, 432444.

[19] J. Han and J Pei, “Mining frequent patterns by patterngrowth: methodology and implications”. ACM SIGKDD Explorations Newsletter 2, 2, 1420. 2000.

[20] G. PiatetskyShapiro. “Discovery, analysis, and presentation of strong rules. Knowledge Discovery in Databases”, 1991: p. 229248.

AUTHORS PROFILE

Yosef Jbara is a lecturer in the Department of Computer Technology at Yanbu College of Technology. He received a Ph.D. in Artificial Intelligence from the Faculty of Computer Information Systems, University of Banking and Financial Sciences. He has published in the areas of artificial intelligence; simulation modeling; data mining and software engineering. His research focuses on artificial intelligence areas as well as simulation and data mining. He has a wealth of expertise gained from his work experiences in Jordan and Saudi Arabia, ranging from systems programming to computer network design and administration.

Marwan Abu-zanona is a lecturer in the Jerash University. He received a Ph.D. in Artificial Intelligence from the Faculty of Computer Information Systems, University of Banking and Financial Sciences. His research interest in neural networks, artificial intelligence and software engineering areas. He has a wealth of expertise gained from his work experiences in Jordan, ranging from web development to network administration.

Farah Al-Zawaideh is the Chairman of Computer Information System in Irbid National University from 2009 until now. He received a Ph.D. in Knowledge based systems from the Faculty of Computer Information Systems, University of Banking and Financial Sciences. His research interest in genetic algorithms, E-learning and software engineering areas. He has a wealth of expertise gained from his work experiences in Jordan, ranging from web development to network administration.