British Journal of Applied Science & Technology
4(22): 3160-3178, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

Efficient Frequent Pattern Mining Using Auto-Associative Memory Neural Network

Suhasini Itkar[1*] and Uday Kulkarni[2]

[1]Department of Computer Engineering, PES Modern College of Engineering, Pune-411005, India.
[2]Department of Computer Science and Engineering, SGGS Institute of Engineering and Technology, Nanded-431606, India.

Authors' contributions

This work was carried out in collaboration between all authors. Author SI designed the study, performed the statistical analysis, wrote the algorithm, and wrote the first draft of the manuscript and managed literature searches. Authors SI and UK managed the analyses of the study and literature searches. All authors read and approved the final manuscript.

Original Research Article

Received 6th April 2014
Accepted 30th May 2014
Published 10th June 2014

ABSTRACT

Aims: Frequent pattern mining is one of the imperative tasks in the data mining. The soft computing techniques such as neural network, fuzzy logic have potential to be used in frequent pattern mining since these powerful tools efficiently model data which is also an essential part of mining. The proposed paper aims to provide efficient mining solution using auto-associative memory neural network to efficiently traverse and reduce the search space, and to reduce the I/O computations. It also aims to keep balance in computational and resource efficiency.
Methodology: This paper proposes an efficient algorithm for mining frequent patterns using auto-associative memory. Auto-associative memory is best suitable artificial neural network (ANN) approach for association rule mining as it stores associations among the patterns. In the proposed system auto-associative memory based on the correlation matrix memory (CMM) is used to find the frequent patterns. The proposed work introduces novel learning and recall algorithms using CMM for mining frequent patterns efficiently. The proposed learning algorithm reduces the search space tremendously for recall mechanism. The proposed recall algorithm uses only frequent 1-patterns and frequent 2-patterns for determining all other frequent patterns, reducing the number of I/O

**Corresponding author: E-mail: suhasini_naik@yahoo.com, kulkarniuv@yahoo.com;*

computations and speeding up the mining process. This approach keeps well balance among computational and resource efficiency.
Conclusion: The performance of the proposed system is compared with traditional algorithms like Apriori, Frequent pattern growth (FP-growth), Compressed FP-tree based algorithm (CT-PRO) and Linear time Closed itemset Miner (LCM). The experimental results show order of magnitude improvement in execution time and storage space optimization to accumulate frequent patterns. Proposed work proves milestone approach in the field of frequent pattern mining using artificial neural network.

Keywords: Association Rule Mining; frequent pattern mining; artificial neural network; auto-associative memory; correlation matrix memory.

1. INTRODUCTION

The data mining techniques can be broadly classified as predictive and descriptive techniques into association rule mining, classification, clustering, and regression and prediction analysis. Data mining is defined as a multi disciplinary research area which integrates various techniques from soft computing field such as machine learning, statistics, ANNs, etc.

Among the descriptive data mining techniques, association rule mining is a two step process. The first step is frequent pattern mining where all frequent patterns are mined using predefined minimum support threshold. In the second step strong association rules are generated from the frequent patterns using minimum confidence threshold. As the second step is less costly in terms of number of computation and is based on the first step, the overall performance of association rule mining is determined mainly by frequent pattern mining.

Introduced by Agrawal et al. [1] association rule mining is a technique to form rules defining the associativity among the attributes of the datasets. Similarly, in the ANNs, the concept of associativity defines an explicit relationship between an input and target patterns offered to the ANN. Earlier less research work appeared using ANN for rule mining may be because of its complex architecture and learning algorithms. Later, rule mining researchers recognized ANN as an important technique.

The rule mining algorithms have been continuously developed by both ANN and non-neural network researchers. In non-neural network based algorithms the Apriori algorithm [1] is a level- wise algorithm where it first process frequent 1-itemsets then frequent 2-itemsets and so on till maximum frequent n-itemsets. Another characteristic of this algorithm is generate-and-test for finding frequent patterns. It requires multiple database scans equal to maximum length of frequent pattern in worst case. It also requires huge memory space to generate and store all these candidate sets.

In neural network based algorithms Craven and Shavilk proposed rule mining as learning for if-then-else rule mining using a trained ANN [2,3]. One approach to tackle the drawback of not gaining accurate interpretations of the nonlinear systems has been developed using methods of rule extraction [4-9]. This approach turns the conversion of the real-valued knowledge formed in the weight matrix into symbolic rules.

The rule extraction mechanisms already exist for different neural architectures such as multilayer perceptron [10-14], recurrent networks [15], self-organizing networks [9,16], associative memory networks [17], genetic algorithms [18,19], Fuzzy rules [20], and even for hybrid architectures [21]. They have focused on the formation of rules, in particular for classification tasks. A Hopfield network for rule mining has already been proposed by Gaber et al. [22], generating association rules from an ANN. Employing incremental training for maintaining rules throughout time, is proposed by Eom and Zhang [23]. Multilevel association rule mining using Multi Level Feed Forward (MLFM) neural network is proposed as a specialized area of rule mining [24,25]. A novel four phase data mining algorithm using ANN, referred as ESRNN (Extraction of Symbolic Rules from ANNs), is proposed for extracting symbolic rules [14]. The algorithm uses back propagation learning.

ANNs have the capability to interpret meaning from complicated data and hence can be used for extracting patterns and detecting trends in various applications like e-commerce applications [26], credit card fraud detection [27], and biomedicine [28].

Other non-neural based algorithms tried to overcome the drawbacks of Apriori algorithm are Frequent pattern growth (FP-growth) [29,30], Compressed FP-tree based algorithm (CT-PRO) [31] and Linear time Closed itemset Miner (LCM) [32].

The FP-growth algorithm proposed compact prefix tree based frequent pattern tree structure FP-tree and the FP-growth algorithm to mine conditional trees. The CT-PRO is another prefix tree based algorithm which utilizes a more dense data structure, Compressed FP-Tree (CFP-Tree). The algorithm achieves better resource efficiency by reducing number of nodes in a CFP-Tree to a half less than in the corresponding FP-Tree, resulting in memory optimization. In LCM it proposed various algorithms like LCM, LCMfreq and LCMmax, for enumerating closed, all and maximal frequent itemsets. It tried to solve the problems of dense datasets. The second version of LCMmax includes a pruning method, thus the computation time is reduced when the number of maximal frequent itemsets is small.

The challenge of the generation of association rules from a dataset is substantial, since its complexity, which is mainly defined by the complexity of frequent pattern mining, is exponential in search space. Although many approaches as discussed above like multilayer perceptron, recurrent networks, Hopfield network and genetic algorithms have claimed reducing the complexity of the problem, the real causes that produce a good performance in the algorithms are still uncertain. Hence, it is important to continue research in this area in order to find answers to the current challenges.

The objective of the proposed work is to implement efficient strategies to traverse and reduce the search space, and to reduce the I/O computations. The proposed auto-associative memory based CMM is best suitable neural network approach for association rule mining as it stores associations among patterns and uses CMM as it provides excellent speed and scalability.

After observing the drawbacks of alternative options available for frequent pattern mining using other neural networks and associative memory approach as discussed in literature review the proposed work based on the auto-associative memory using CMM has been carried out. All the type of associative memories uses CMM structure as it deals with binary and non-binary data and has fast training and recall stages as defined by Austin and Stonham in 1987 and Haykin in 1999 and uses hebbian learning rule to train the CMM as it provides excellent speed and scalability advantage as stated by Haykin.

An auto-associative memory using CMM structure is able to learn, store and recall itemset support after it have been trained with data defining associations. Especially, we analyze and decode the training process and the CMM weight matrix of an auto-associative memory to propose itemset support extraction mechanisms through which they are able to recall itemset support when an itemset is presented as stimulus to the trained networks.

The novel learning and recall algorithms are proposed for performing these tasks efficiently. The proposed learning algorithm reduces the search space tremendously for recall mechanism. The proposed recall algorithm uses only frequent 1-patterns and frequent 2-patterns for determining all other frequent patterns, reducing the number of I/O computations and speeding up the mining process. The input key parameters are $\min\_sup$ and $\min\_conf$ and output key parameter is mining time.

The remainder of this paper is organized as follows: Section 2 reviews work related to frequent pattern mining using ANN approaches. Section 3 describes preliminaries for auto-associative memory. Section 4 introduces a novel learning and recall algorithm using CMM for mining frequent patterns efficiently. Section 5 shows experimental results and comparison of proposed implementation with Apriori, FP-growth, CT-PRO and LCM algorithms. Finally, Section 6 draws conclusions from the proposed work and discuss about future scope.

2. RELATED WORK

This section represents the problem statement and related work done in the area of frequent pattern mining using ANN.

2.1 Problem Definition

Let any element i from a set $I = \{i_1, i_2, .., i_m\}$ be called an *item* and a grouping or formation of items X such that $X \subseteq I$ be called an *itemset*. In particular, an itemset X with $k = |X|$ is called a *k*-itemset. Let database D be a set of n transactions, representing an environment. Each transaction t_i is defined by a unique identifier *id* together with an itemset Y, satisfying $Y \subseteq I$. A t_i is said to support an itemset X if $X \subseteq t_i.Y$.

A basic association rule is an implication defined by $X \rightarrow Y$ in which the itemsets $X \wedge Y \subset I$, but $X \cap Y = \emptyset$.

The support $supp(X)$ of an itemset X with respect to D is defined by the fraction of transactions supporting it. This can be defined as follows:

$$supp(X) = P(X) = \frac{|\{t_i | t_i \in D \wedge t_i.Y \supseteq X\}|}{|D|} = \frac{freq(X)}{n} \quad (1)$$

Since the $supp(X)$ defines the occurrence frequency of X in D, it can also be understood as the probability of X , $P(X)$. In the case of a rule defined by $X \rightarrow Y$, its support is given by the support of $X \cup Y$. The strength of a rule in D is defined by its confidence $conf(X \cup Y)$ as follows:

$$conf(X \cup Y) = \frac{supp(X \cup Y)}{supp(X)} \quad (2)$$

Since it is impractical and not desirable to mine and generate the total space of itemsets and rules, the challenge of rule mining has been to discover only those rules which are interesting. The interestingness of an itemset and a rule is determined by evaluating the properties of support and confidence against the thresholds of minimum support threshold, min_sup ξ and minimum confidence threshold, min_conf respectively. For instance, an itemset X is *frequent* iff $supp(X) \geq \min\_\sup$ ξ.

Most association rule mining algorithms employ a support-confidence framework. This framework exclude the good number of uninteresting rules but the bottleneck of this approach is filtering uninteresting rules especially when mining at low support thresholds or mining for the long patterns. To tackle this weakness, a correlation measure can be used like lift, χ^2, all_confidence, max_confidence, Kulczynski and cosine measures.

2.2 Literature Review

Frequent Pattern Mining using ANN is less researched area. The techniques like associative memories, feed forward neural network, and competitive neural network like self organizing map are already explored.

Dong et al. [13] proposed competitive network based FP-growth (CNFP) method for mining frequent patterns. This method combines competitive neural network with FP-growth. In competitive learning grouping of similar patterns is carried out by the network and represented by a single neuron. Grouping of similar patterns is done automatically based on data correlations and large dataset is divided into sets of similar patterns. After competitive learning, to construct FP-sub-trees neurons in competitive layer are considered as root and FP-sub-trees contain transactions similar to each other. Frequent patterns are mined using FP-sub-tree which in turn reduces the search space, whereas for large datasets competition phase may take more time for generating neurons.

The paper [14] proposed a novel four-phase data mining algorithm using ANNs, referred as Extraction of Symbolic Rules from ANNs (ESRNN), for extracting symbolic rules. The algorithm uses back propagation learning. By using heuristic clustering algorithm, the nodes in the hidden layers are discretized. Then symbolic rules are extracted from frequent patterns using extraction algorithm. An important feature of the proposed rule extraction algorithm is its recursive nature.

The Vicente Oswaldo [17] proposed algorithms for frequent pattern mining using auto-associative neural network and the self-organizing map. In this paper the learning and recall algorithms are introduced to mine frequent patterns. The neural-based proposed framework involves the building of artificial memories that are able to learn, store and recall itemset support. In learning process the weight matrix of an itemset support is constructed based on the associations among patterns. In recall phase frequent patterns are mined. It utilizes inherent feature of auto-associative memory to store associations among patterns. However, the case of finding frequent patterns for an itemsets larger than two items remains an open problem.

Various soft computing tools are proposed using real life applications like Long-term prediction of discharges in Man wan Reservoir, analysis of outcomes of construction claims, Swarm Optimization [33-38]. The proposed work can be extended further for using real life applications in the field of web mining, text mining and bioinformatics.

The scope of the research work is to propose a novel and efficient algorithm to overcome the drawbacks of previously proposed approaches and utilize inherent features of auto-associative memory more efficiently.

3. AUTO-ASSOCIATIVE MEMORY BASICS

The concept of an auto-associative memory is mainly exploited by neural architectures which imitate the concept of associative memory present in our brain. In general terms, the explicit association among the inputs is the target to be learnt by some networks to form a knowledge which can be used for pattern association or recognition tasks.

The frequent pattern association involves associating a new pattern with a stored pattern. The associative memory is considered as a special case of the neural computing approach for pattern recognition. Types of associative memory are the Hetero-Associative memory, an Auto-Associative memory, the Hopfield Network and the Bidirectional Associative Memory (BAM).

The Hopfield network for association rule mining has been proposed by Gaber et al. [22] but is used to determine maximal itemsets so the main drawback of this work is that to calculate support for all itemsets derived from the maximal discovered itemsets an extra pass over the training data is needed.

After observing the drawbacks of alternative options available for frequent pattern mining using associative memory the proposed work based on the auto-associative memory using CMM has been proposed.

The auto-associative memory is the best suited neural network approach for association rule mining as it stores associations among patterns. In the proposed system auto-associative memory based on CMM is used to find frequent patterns. Associative memory can be feed forward or recurrent. This is supervised single-layer network.

3.1 Architecture of Auto-Associative Memory

The architecture of an Auto-Associative Memory is shown in the Fig. 1. It is based on the concept of association.

The auto-associative memory neural network is first trained to store a set of patterns in the form $X : Y$ where X represents the input vector and Y is the corresponding output associative vector. The input patterns are represented as unipolar vectors $\in \{0,1\}$. It has the property to remember the associativity expressed between input patterns under the supervised training.

Since it has been stated by Haykin in 1999 that the target y_i pattern will be defined by the corresponding key pattern x_i of each input pair, proposed study could be confined to the study of the suitability of an auto-associative memory for frequent pattern mining since this neural network holds the charactseristic of $y_i = x_i \; \forall \; x_i$ in D in its inputs.

It uses the CMM to store the associations among patterns. The CMM is a single-layer memory structure defined as a square matrix whose dimension is the $m \times m$ elements. The CMM structure is used to store frequencies or pattern occurrences in the form of weights.

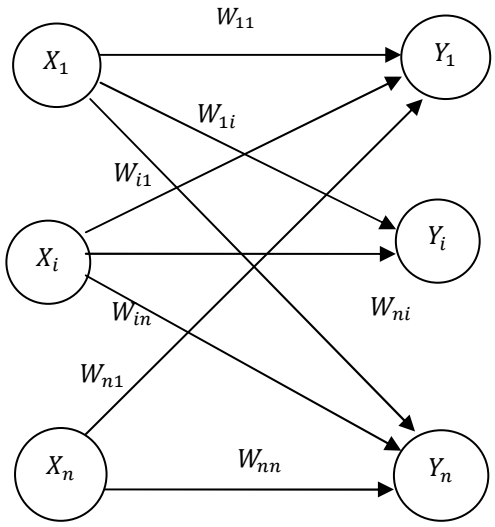

Fig. 1. Auto-associative memory

The CMM Models are binary weighted and non-binary weightless and has fast training and recall stages. The CMM resulting from the supervised training is represented as $M \in B^{m \times m}$ for binary weighted model. The CMM is trained using a hebbian learning rule which offers excellent speed and scalability advantages. The work utilizes inherent features of CMM and proposes an efficient algorithm for frequent pattern mining.

4. FREQUENT PATTERN MINING USING AUTO-ASSOCIATIVE MEMORY

In order to perform frequent pattern mining with an auto-associative memory using CMM, focus is on how to perform the counting of patterns or itemsets. It is normally produced by looking for the patterns by scanning the high dimensional space defined by the original data environment.

In the work learning algorithm is proposed first using weighted CMM matrix, where it accomplishes the task of training the neural network. The outcome of a learning algorithm is the trained weighted CMM matrix where weights represent the occurrence frequency of associations among patterns. The efficiency for this algorithm is achieved by optimizing the search space as explained in section 4.1.

Then the proposed recall algorithm applies a novel approach to recall itemset support from the trained CMM matrix. The proposed recall algorithm mine frequent patterns efficiently using weighted CMM matrix, hence reduces the I/O computations as explained in section 4.2.

The following sections describe a two step novel frequent pattern mining algorithm using CMM matrix memory.

4.1 Learning for Itemset Support Using CMM

The task of itemset support counting is mainly governed by the learning algorithm responsible for modifying the neural-network architecture (nodes or weight matrix). It accumulates the knowledge presented in terms of support counts in the environment throughout the time. Therefore, the learning algorithm is primarily determined by the type of neural network used to learn the data coming from the environment. In our particular case, the learning algorithm of an auto-associative memory will be evaluated to fulfill this framework task.

The binary weighted CMM matrix M represented as $M \in B^{m \times m}$ is used to store frequencies or pattern occurrences in the form of weights. Learning is done using the natural ability of the CMM to learn about occurrence frequency of the patterns based on input patterns.

Input to this step is database D represented as a group of n unipolar vectors $\in \{0,1\}$, where each input is an m-vector defining a particular association among items from the set $I = \{i_1, i_2, .., i_m\}$ where m represents the total number of items and transactions from the set $T = \{t_1, t_2, \ldots., t_n\}$ where n is total number of transactions in database D.

Training requires pairs of patterns representing associations with a form $X \rightarrow Y$ in which an input or key pattern is associated with an output or memorized pattern. Corresponding associations among patterns may look as,$\{(X_1, Y_1), (X_2, Y_2), \ldots., (X_n, Y_n)\}$, where both the patterns, the input and the output, will obtain their values from the n transactions in D. In particular, each pattern of each pair will take the same value defined by a transaction t_k. Hence pair k satisfies $X_k \rightarrow Y_k = (X_k, Y_k)$ and represented as $X_k = Y_k = t_k.x$. The learning algorithm is proposed as described below which takes input as pairs of patterns and output is weighted CMM M.

Learning Algorithm:

- To train M, the equation defined (Haykin, 1999; Ham and Kostanic, 2001) is as in (3).

$$M = \sum_{k=1}^{m} Y_k X_k^T \tag{3}$$

- The term $Y_k X_k^T$ is an estimation of the weight matrix $W(k)$ of the neural network functioning as a linear associator. This matrix $W(k)$, which associates X_k to Y_k, forms a mapping representing the association described by the k^{th} input pair in turn.
- The resulting matrix M is grouping of the m weighted matrices W.
- Individual weight values of the network, whose update resembles a generalization of the Hebbian rule learning, can be expressed by (4).

$$w_{ij} = \sum_{k=1}^{m} y_{ik} x_{jk} \tag{4}$$

- As a consequence of using unipolar elements as inputs, the product $y_i x_j$ of some k^{th} pattern will be in one of the two states as shown in (5).

$$y_{ik} x_{jk} = \begin{cases} 1 & \text{existence of association } ij \text{ in } k \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The output of symmetric M, from which itemset support values will be estimated, is represented in (6)

$$M = \begin{bmatrix} w_{11} & & 0 \\ w_{21} & w_{22} & \\ \vdots & \vdots & \ddots \\ w_{m1} & w_{m2} & \dots \; w_{mm} \end{bmatrix} \quad (6)$$

In the weighted CMM M, w_{ij} not only means the existence of an association between elements i and j, but also the number of times that such association has occurred in the environment. Therefore, we can asseverate that a weighted symmetric CMM naturally builds a frequency matrix which is used in recall algorithm to mine frequent patterns efficiently.

Advantage of learning algorithm is it optimizes space required to store frequent 1-patterns and frequent 2-patterns in weighted CMM matrix. It also reduces search space for recall mechanism to focus on the knowledge of only $n(n+1)/2$ nodes rather than the n^2 elements of the complete matrix.

4.2 Recall for Finding Frequent Patterns Using CMM

As a result of learning, pattern frequency is embedded in the knowledge of a neural network; the focus of this section is on giving the right interpretation to the weight matrix in order to recall the frequency (support) of patterns, which can be composed by different items of the learnt environment.

To use the embedded information in triangular M matrix, the right interpretation needs to be drawn to produce accurate support values from it. Therefore, to achieve accurate itemset support recalls from M, when a stimulus is presented, a novel recall algorithm is proposed. The output of recall algorithm is mined frequent 1-itemsets, 2-itemsets, … , n-itemsets respectively.

Algorithm Recall (weighted CMM M , int min_sup ξ)

- Determine the support of all items $I = \{i_{1,} i_{2,} \ldots, i_m\}$ stored in M, applying equation (7), where $freq(w_{ii})$ is occurrence frequency of pattern i stored in M at index ii. Support for pattern i is determined as $P(w_{ii})$ which is the probability of the i^{th} item in the n transactions defining the training dataset.

$$supp(i) = P(w_{ii}) = \frac{freq(w_{ii})}{n} \quad (7)$$

  If $supp(i) \geq \xi$ then that pattern is stored as frequent 1-itemset or frequent 1-pattern.

- Determine the frequent 2-itemsets (ij) applying formula as in (8) to all m items stored in M.

$$supp(ij) = P(w_{ij}) \quad = \frac{freq(w_{ij})}{n} \quad for\ all\ i \neq j \tag{8}$$

If $supp(ij) \geq \xi$, then pattern (ij) is stored as frequent 2-itemset.

- Determine support for k-itemsets $(2 > k \leq m)$ using following steps-

    a. Only frequent 1-itemsets and frequent 2-itemsets are considered as per antimonotone property (if subset of any pattern is infrequent then its superset are also infrequent)
    b. Determine frequent k-itemsets applying equation as in (9) to combinations of only frequent 1-itemsets and frequent 2-itemsets stored in M.

$$P(X) = \frac{\sum_{i=1}^{k} \sum_{j=i}^{k} P(w_{ij})}{n} \tag{9}$$

Where, all unique pairs of X_i i.e., $n = {}^{k}C_2$ and k is length of itemset.

Proposed novel recall algorithm is more efficient as it finds k-itemsets $(2 > k \leq m)$ using frequent 1-itemsets and frequent 2-itemsets and ignores all infrequent patterns. Another advantage is it reduces I/O cost as there is no need to scan original database various times to find frequent patterns as recall is carried out using knowledge embedded in weighted CMM matrix M.

As an example consider database DB as shown in Table 1, with 5 transactions. Let minimum absolute support threshold ξ is 3 and relative support threshold on the scale of 10 to 100 is 60% and on the scale of 0 to 1 is 0.6.

Table 1. The transaction database

Transaction ID	Items Input
1001	3, 6, 1, 4, 7, 13, 16, 9
1002	3, 6, 1, 2, 13, 12, 15
1003	6, 2, 8, 10, 15
1004	3, 16, 2, 11, 19
1005	3, 6, 1, 13, 16, 5, 12, 14

The output M of learning algorithm for example is as shown below where CMM M is of size 19×19, as maximum number of items in itemset are 19. The learning algorithm generates symmetric weighted CMM M. It trains matrix M with frequency count of all 1-itemsets and 2-itemsets in one single database scan with supervised training and speed up the mining process.

Showing weighted CMM M

$$\begin{matrix}
3&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&3&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
3&2&4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
3&2&3&1&1&4&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{matrix}$$

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

The recall algorithm uses weighted CMM M to generate frequent 1-itemsets and frequent 2-itemsets efficiently with the advantage of reduced search space. It discards all infrequent 1-itemsets and 2-itemsets. Taking antimonotone property into consideration all k-itemsets $(2 > k \leq m)$ are mined using frequent 1-itemsets and frequent 2-itemsets, hence speed up task of mining. The outputs of Recall algorithm for given example for certain cases are as shown below:

- Case 1: for $k = 3$, therefore ${}^{3}C_{2}$=3 so $X = (X_1, X_2, X_3)$. Then, to find out whether pattern (1 2 3) is frequent or not recall equation is used.

$$P(1\ 2\ 3) = \frac{P(w_{2\,1}) + P(w_{3\,1}) + P(w_{3\,2})}{3}$$

$$P(1\ 2\ 3) = \frac{\frac{1}{5} + \frac{3}{5} + \frac{2}{5}}{3} = 0.4$$

As $P(1\ 2\ 3) = 0.4 < 0.6$ (ξ), this pattern is infrequent 3 - itemset.

- Case 2: for k = 4, therefore ${}^{4}C_{2}$ = 6 so X = (X_1 , X_2 , X_3 , X_4). Then, to find out whether pattern (1 3 6 13) is frequent or not recall equation is used.

$$P(1\ 3\ 6\ 13) = \frac{P(w_{3\,1}) + P(w_{6\,1}) + P(w_{13\,1}) + P(w_{6\,3}) + P(w_{13\,3}) + P(w_{13\,6})}{6}$$

$$P(1\ 3\ 6\ 13) = \frac{\frac{3}{5} + \frac{3}{5} + \frac{3}{5} + \frac{3}{5} + \frac{3}{5} + \frac{3}{5}}{6} = 0.6$$

As $P(1\ 3\ 6\ 13) = 0.6 = 0.6$ (ξ), this pattern is frequent 4 - itemset.

The final output of recall algorithm for given example is:

Generating candidates: 1 – itemset

Possibilities: 19

{1}: 3.0: 0.6
{2}: 3.0: 0.6
{3}: 4.0: 0.8
{6}: 4.0: 0.8
{13}: 3.0: 0.6
{16}: 3.0: 0.6

Generating candidates: 2 – itemset

Possibilities: 171

{1 3}: 3.0: 0.6
{1 6}: 3.0: 0.6
{1 13}: 3.0: 0.6
{3 6}: 3.0: 0.6
{3 13}: 3.0: 0.6
{3 16}: 3.0: 0.6
{6 13}: 3.0: 0.6

Generating candidates: 3 – itemset

Possibilities: 969

{1 3 6}: 3.0: 0.6
{1 3 13}: 3.0: 0.6
{1 6 13}: 3.0: 0.6
{3 6 13}: 3.0: 0.6

Generating candidates: 4 – itemset

Possibilities: 3876

{1 3 6 13}: 3.0: 0.6

Generating candidates: 5 - itemset
Possibilities: 11628

Time required finding frequent candidate is: 0.015 seconds

If the results are compared for a given example, time required to mine frequent patterns is less than the standard frequent pattern mining algorithms like Apriori, FP-growth, CT-PRO and LCM algorithms.

5. EXPERIMENTAL RESULTS

This section will provide the experimental results of the proposed system. The key input parameter of proposed algorithm is minimum support threshold ξ and the output parameters considered for comparison are time required for finding frequent 1-itemsets (time to learn the neural network) and total execution time (Time to learn and recall). For lower ξ values mining requires more execution time as it generates large number of frequent patterns and higher the ξ range lesser the mining time.

The proposed work at present is implemented based on support-confidence framework and can be further extended using correlation measure like Kulczynski measure, together with the imbalance ratio, to discover the quality of the patterns.

In order to determine the accuracy of the proposed mechanisms for itemset support recalls from a trained CMM, experiments with standard real datasets are performed in this section. The experiments were performed on an Intel Core2 Duo processor 1.66GHz with 2GB memory, running the Microsoft Windows.

The proposed system is compared with non-neural network based algorithms as it is observed that very few neural network based algorithms compared their work with standard and large datasets. The major achievement of our proposed work is in neural network based algorithms, it proves good mining efficiency on standard datasets where other algorithms failed to prove their work. So this algorithm proves a milestone in the research of frequent pattern mining using neural network. This work is also extended for handling huge datasets using Hadoop MapReduce framework and is not included in the scope of this paper.

The standard datasets used in previous studies are downloaded from Frequent Itemset Mining Implementations Repository (FIMI'04) website [39] and used to compare the algorithms performance with non-neural network based algorithms like Apriori, FP-growth, CT-PRO and LCM algorithms. Experiments are performed on various dense and sparse datasets from FIMI Repository. The experimental results of the proposed work are consistent for all the datasets while compared with non-neural network based algorithms.

Mushroom dataset is a sparse dataset with 8124 transactions, 119 items and average transaction length 23, hence it is relatively sparse. Time required for finding frequent 1-itemsets using mushroom dataset is shown in Table 2 for Apriori and proposed frequent pattern mining with an auto-associative memory using CMM algorithms. It proves that frequent 1-itemsets generation time efficiency obtained by CMM in real datasets like mushroom is also notable with respect to various minimum support thresholds.

Table 2. Time required for finding frequent 1-itemsets

Algorithm/ Min. Support	10%	20%	30%	40%	50%	60%	70%	80%	90%
Apriori Time Taken (Sec)	0.422	0.422	0.437	0.453	0.422	0.406	0.438	0.421	0.422
CMM Time Taken (Sec)	0.171	0.156	0.156	0.141	0.109	0.093	0.093	0.093	0.094

Time required for training, recall phase and total mining time for CMM using mushroom dataset is shown in Table 3 for minimum support threshold range of 50 to 90%. Proposed learning algorithm reduces search space for recall mechanism and achieves resource efficiency by optimizing search space. Proposed recall algorithm uses only frequent 1-patterns and frequent 2-patterns for calculating $(2 > k \leq m)$ itemsets, improving the computational efficiency.

Table 3. Time required for mining frequent patterns using CMM for mushroom dataset

Min. Support	50%	55%	60%	65%	70%	75%	80%	85%	90%
Time to train(sec)	0.109	0.094	0.093	0.093	0.093	0.094	0.093	0.11	0.094
Time to recall (Sec)	0.235	0.062	0.032	0.016	0.079	0.015	0.016	0.031	0.047
Total Time (Sec)	0.375	0.203	0.172	0.156	0.172	0.156	0.156	0.141	0.141

Table 4 shows comparison for mushroom dataset based on mining time for Apriori, FP-growth, LCM, CT-PRO and CMM algorithms. The results of all algorithms are compared for the minimum support threshold range of 50 to 90%.

Table 4. Comparison of Mining time for mushroom dataset

Algorithm/Min. Support	50%	55%	60%	65%	70%	75%	80%	85%	90%
Apriori Time (Sec)	1.812	1.406	1.094	1.015	0.953	0.938	0.969	0.797	0.641
CMM Time (Sec)	0.375	0.203	0.172	0.156	0.172	0.156	0.156	0.141	0.141
LCM Time (Sec)	0.5	0.4	0.317	0.252	0.23	0.225	0.225	0.187	0.187
CT-PRO Time (Sec)	0.55	0.42	0.4	0.4	0.38	0.35	0.3	0.25	0.2
FP-growth Time (Sec)	0.7	0.6	0.4	0.4	0.4	0.3	0.3	0.3	0.3

As shown in Fig. 2 the proposed CMM algorithm shows better efficiency as compared to Apriori, FP-growth, CT-PRO and LCM algorithms.

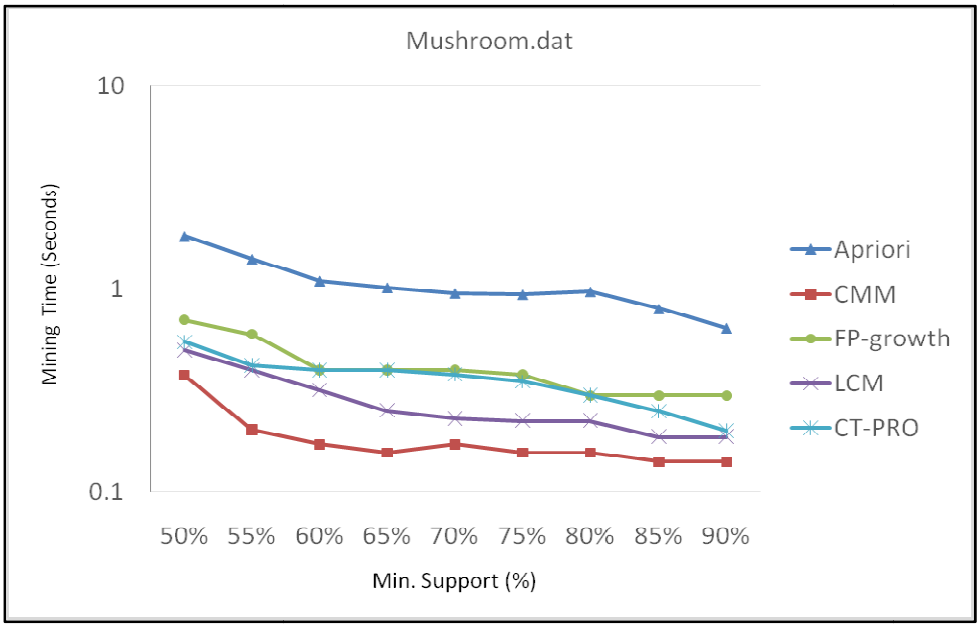

Fig. 2. Mining time for mushroom dataset

For chess dataset proposed CMM algorithm achieves good mining time efficiency. Chess dataset contain 3196 transactions, 75 items and average transaction length is 37, hence it is denser. Although being dense it takes comparatively lesser time for mining than Apriori, FP-growth and CT-PRO algorithms and is slightly comparable with LCM algorithm. Comparative analysis of mining time for chess dataset is as shown in Fig. 3.

The results obtained using the large dataset, accidents, with respect to the various minimum support thresholds is shown in Fig. 4. This dataset is relatively large than the other datasets with 340183 transactions, 468 items and average transaction length 34, hence it is relatively

sparse. Results prove the efficiency of mining large datasets as compared to FP-growth and LCM algorithms.

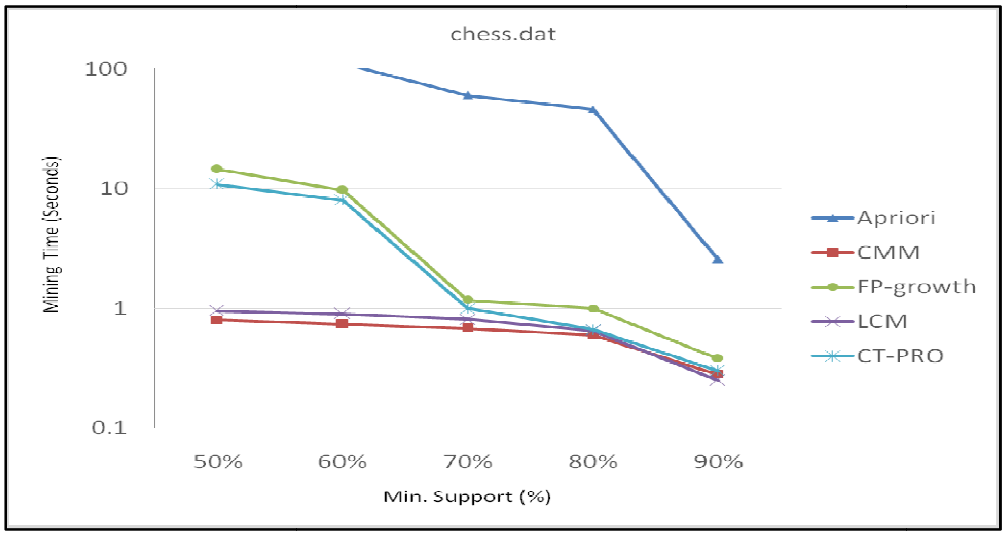

Fig. 3. Mining time for chess dataset

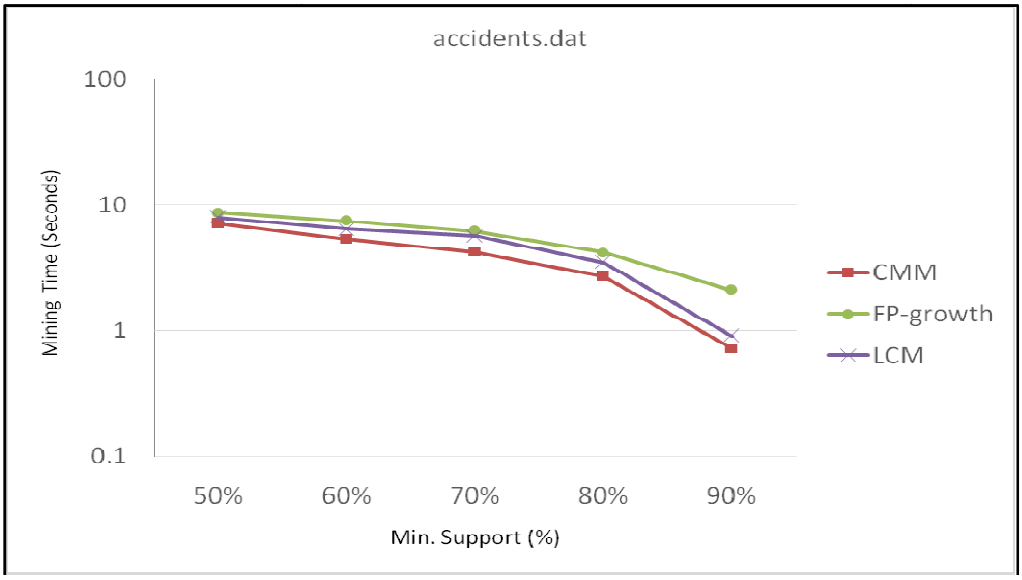

Fig. 4. Mining time for accidents dataset

As shown in Fig. 5, BMS-WebView-2 dataset is tested and compared using CMM, FP-growth and LCM algorithms. The implementation results of CMM shows better computational efficiency as compared to FP-growth and LCM.

These experimental results prove efficiency of the proposed CMM algorithm on sparse and dense datasets. The proposed CMM algorithm even work efficiently for low minimum supports.

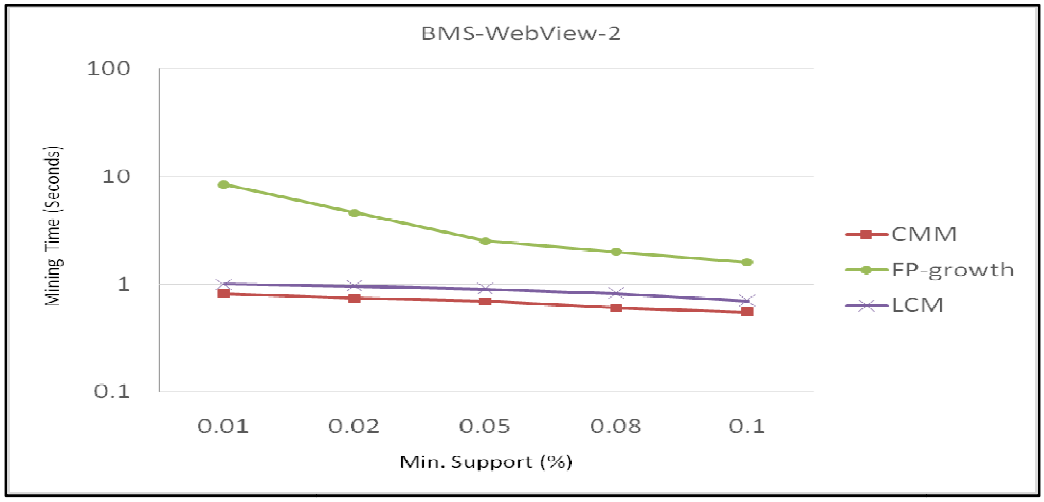

Fig. 5. Mining time for BMS-WebView-2 dataset

For improving mining efficiency of dense datasets and for mining huge and voluminous datasets useful in Big Data analysis, the proposed algorithm is further extended using distributed processing on Hadoop MapReduce environment but is not the scope of this paper.

6. CONCLUSION

Auto-Associative memory based CMM is best suitable neural network approach for association rule mining due to its inherent capability to store associations among patterns. Proposed learning algorithm reduces search space for recall mechanism to focus on the knowledge of only $n(n+1)/2$ nodes rather than the n^2 elements of the complete matrix. Proposed recall algorithm uses only frequent 1-patterns and frequent 2-patterns for calculating $(2 > k \leq m)$ itemsets, speeding up mining process.

This approach keeps well balance among computational and resource efficiency. The major achievement of our proposed work is in neural network based algorithms, it proves good mining efficiency on standard datasets where other neural network based algorithms failed to prove their work. Proposed work proves milestone approach in the field of frequent pattern mining using artificial neural network.

For improving mining efficiency for mining huge and voluminous datasets useful in Big Data analysis, the proposed algorithm is further extended using distributed processing on Hadoop Map Reduce environment but is not the scope of this paper. Future scope of the research is in advanced frequent mining like sequential pattern mining, closed and maximal frequent pattern mining and distributed pattern mining.

COMPETING INTERESTS

Authors declare that there are no competing interests.

REFERENCES

1. Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. Proc. ACM-SIGMOD international conference on management of data (SIGMOD'93), Washington, DC. 1993;207–216.
2. Towell GG, Shavlik JW. Extracting refined rules from knowledge-based neural networks. Machine Learning. 1993;13:71–101.
3. Craven MW, Shavlik JW. Using neural networks for data mining. Future Generation Computer Systems. 1997;13(2–3):211–229.
4. Benitez JM, Castro JL, Requena I. Are artificial neural networks black boxes? IEEE Transactions on Neural Networks. 1997;8(5):1156–1164.
5. Tickle AB, Andrews R, Golea M, Diederich J. The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. IEEE-NN. 1998;9(6):1057.
6. Taha IA, Ghosh J. Symbolic interpretation of artificial neural networks. Knowledge and Data Engineering. 1999;11(3):448–463.
7. Tsukimoto H. Extracting rules from trained neural networks. IEEE-NN. 2000;11(2):377.
8. Browne A, Hudson B. Knowledge extraction from neural networks. Proceedings of the 29th Annual Conference of the IEEE Industrial Electronics Society, Roanoke, Virginia, USA. 2003;1909–1913.
9. Malone J, McGarry K, Wermter S, Bowerman C. Data mining using rule extraction from kohonen self-organising maps. Neural Computing and Applications. 2006;15(1):9–17.
10. Duch W, Adamczak R, Grabczewski K. Extraction of logical rules from training data using backpropagation networks. First Polish Conference on Theory and Applications of Artificial Intelligence. 1996;210.
11. McGarry KJ, Wermter S, MacIntyre J. Knowledge extraction from radial basis function networks and multi-layer perceptrons. IEEE International Conference on Neural Networks (IJCNN'99), Washington DC. 1999;4:2494–249.
12. Setiono R. Extracting M-of-N rules from trained neural networks. IEEE-NN. 2000;11(2):512.
13. Yihong Dong, Xiaoying Tai, Jieyu Zhao. A Distributed Algorithm Based on Competitive Neural Network for Mining Frequent Patterns. International Conference on Neural Networks and Brain. 2005;1:499- 503.
14. Kamruzzaman SM, Jehad Sarkar AM. A New Data Mining Scheme Using Artificial Neural Networks. Sensors. 2011;11:4622-4647. doi:10.3390/s110504622. Jacobsson H. Rule Extraction from recurrent neural networks: A taxonomy and review. Neural Computing. 2005;17(6):1223–1263.
15. Malone J, Mc Garry K, Wermter S, Bowerman C. Data mining using rule extraction from kohonen self-organising maps. Neural Computing and Applications. 2006;15(1):9–17.
16. Hammer B, Rechtien A, Strickert M, Villmann T. Rule extraction from self-organizing networks. Dorronsoro JR editor, 213 ICANN, Lecture Notes in Computer Science. 2002;2415:877–883.

17. Vicente Oswaldo Baez Monroy and Simon O'Keefe. The Identification and Extraction of Itemset Support defined by the Weight Matrix of a Self-Organising Map. IEEE World Congress on Computational Intelligence; 2006.
18. Markowska U, Wnuk P. Rule Extraction from Neural Network by Genetic Algorithm with Pareto Optimization. Springer Lecture Notes in Computer Science. 2004;3070(1):450-455.
19. Peter Wakabi-Waiswa, Venansius Baryamureeba, K Sarukesi. Generalized Association Rule Mining Using Genetic Algorithms. Computer Science, Proceedings of Seventh International conference in natural computation, IEEE. 2011;59-69.
20. Au W, Chan K. Mining Fuzzy Association Rules. Proc. of the 6th international conference on Information and knowledge management, Las Vegas, Nevada, USA. 1997;209-215.
21. Eggermont J. Rule-extraction and learning in the BP-SOM architecture. Leiden university, Internal Report IR-98-16 (Masters Thesis), August; 1998.
22. Gaber K, Bahi J, El-Ghazawi T. Parallel mining of association rules with a hopfield type neural network. 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2000), Vancouver, Canada. 2000;90–93.
23. Eom JH, Zhang BT. Prediction of east protein-protein interactions by neural feature association rule; 2005.
24. Tong-yan Li, Xing-ming Li. Study of the neural network applied to weighted Association Rules Mining. Proc. International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China. 2-4 Nov; 2007.
25. Amit Bhagat, Sanjay Sharma, Pardasani KR. Ontological Frequent Patterns Mining by potential use of Neural Network. International Journal of Computer Applications (0975–8887). 2011;36(10):44-53
26. Pao-Hua Chou, Pi-Hsiang Li, Kuang-Ku Chen, Menq-Jiun Wua. Integrating web mining and neural network for personalized e-commerce automatic service. Expert Systems with Applications. 2010;37:2898–2910.
27. Baesens B, Setiono R, Mues C, Vanthienen J. Using neural network rule extraction and decision tables for credit risk evaluation. Management Science. 2003;49(3):312-329.
28. Robardet C, Cremilleux B, Boulicaut J. Characterization of Unsupervised Clusters with the Simplest Association Rules: Application for Child's Meningitis. Seventh International Workshop on Intelligent Data Analysis in Biomedicine and Pharmacology (IDAMAP'02), Lyon, France. 2002;61-66.
29. Han J, Pei J, Yin. Mining frequent patterns without candidate generation, In: Proceedings of the ACM-SIGMOD conference management of data. 2000;29(2):1–12.
30. Han J, Pei J, Yin Y, Mao R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach, Data Mining and Knowledge Discovery. 2004;8(1):53–87.
31. Sucahyo, Yudho Giri, Gopalan P. Raj. CT-PRO: A Bottom-Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure. IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI). 2004;126.
32. Takeaki Uno, Masashi Kiyomi, Hiroki Arimura. LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets. In: IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI). 2004;126:1-11.
33. Huang ZK, et al. A New Image Thresholding Method Based on Gaussian Mixture Model. Applied Mathematics and Computation. 2008;2:899-907.
34. Taormina R, et al. Artificial Neural Network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. Engineering Applications of Artificial Intelligence. 2012;25(8):1670-1676.

35. Wu CL, et al. Predicting monthly streamflow using data-driven models coupled with data preprocessing techniques. Water Resources Research 45, W08432, doi: 10.1029/ 2007WR006737; 2009.
36. Zhang J, et al. Multilayer Ensemble Pruning via Novel Multi-sub-swarm Particle Swarm Optimization. Journal of Universal Computer Science. 2009;15(4):840-858.
37. Cheng CT, et al. Long-term prediction of discharges in Manwan Reservoir using artificial neural network models. Lecture Notes in Computer Science. 2005;3498:1040-1045.
38. Chau KW. Application of a PSO-based neural network in analysis of outcomes of construction claims. Automation in Construction. 2007;16(5):642-646.
39. Frequent Itemset Mining Implementations Repository (FIMI) Repository; 2004. Accessed on 1 June 2013. Available: http://fimi.cs.helsinki.fi.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history.php?iid=551&id=5&aid=4863