MADPARM: Mobile Agent based Distributed and Parallel Association Rule Mining

A.Saleem Raja [#1], E.George Dharma Prakash Raja [#2]
[#1] *Research Scholar,*
[#2] *Assistant Professor,*
Department of Computer Science, Engineering and Technology,
Bharathidasan University, Trichy, Tamil Nadu,
India.

Abstract *Distributed and Parallel data mining requires flexible and extensible framework to mining the useful knowledge from distributed database sites. Frequent Item-set mining is the primary step in association rule mining. Plenty of research work had been done in the distributed data mining especially association rule mining. Recently researcher deployed mobile agents for distributed association rule mining (IDMA, AEDM, AMAARM, EMADS, MADM , AeMSAR and MAD-ARM). Most of the approaches focused on reducing the communication cost by deploying multiple mobile agents and establish protocol for communication between them. This paper introduces the new novel framework which improves MADARM with parallel access of data from distributed site's database using mobile agents (MADPARM). To improve the performance, we used compact bit table approach for mining frequent item-set (FI) from each site. Finally we present the result which shows the proposed framework gives better result than the MADARM in distributed environment.*

Keywords*: Distributed Association Rule Mining, Frequent Item Set Mining, Parallel Mining, Mobile agent based distributed FI mining. CBT-FI based distributed association rule mining, Compact BitTable based FI*

1 INTRODUCTION

Ultimate goal of data mining is to discover potentially useful information embedded in databases. Distributed nature of business databases paves the way of distributed and parallel data mining. Large scale distributed and parallel data mining, requires intelligent miner which can adopt for different mining strategy at each site and integrates the result seamlessly. This requirement integrates autonomous agent with distributed data mining. Agent can be treated as a computing unit that performs multiple tasks based on a dynamic configuration [1]. Integration of distributed data mining with mobile agent creates the new research direction and initiates several research problems in distributed data mining[20] such as reducing communication cost, handling multiple heterogeneous data sources, efficiency of incremental knowledge integration, scalability of the framework, data privacy & security, mobile agent security, fault tolerance and efficient data partitioning. This paper we attempt to improve speed of FI generation in distributed sites by using CBT-fi algorithm [11] and gathering FI from distributed sites using mobile agent framework MADPARM. MADPARM is the improved form of MADARM framework [19] with parallel accessing of distributed sites. MADPARM uses less number of agents to communication overhead between agents.

Rest of the paper is organized as follows. Section 2 presents the overview of Association Rule Mining. Section 3 present the review of existing bit table based frequent itemset mining. The Overview of the mobile agents and existing agent based framework for association rule mining are discussed in section 4. Proposed framework is explained in section 5. Section 6 presents the experimentation and analysis. Finally we conclude the paper in section 7.

2 ASSOCIATION RULE MINING

Association rule mining is one of the data mining techniques, introduced by Agrawal et al in 1993 [2]. It finds the interesting association and/or correlation relationships among large set of data items. Discovering this association rules can guide the decision making. Association Rule Mining includes two major steps such as frequent item-sets (FI) mining and strong association rule generation. But complexity of FI mining is significantly greater than that of association rule generation. A typical and widely used example of frequent item-sets mining is to analyze supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Frequent sets of products describe how often items are purchased together. In addition to this frequent item-sets mining have applications in areas such as bioinformatics, fraud detection and web usage mining [3]. FIM algorithms generally classified into two types, candidate generation and pattern growth.

- Candidate generation algorithms (e.g. Apriori [2]) generates candidates based on previously identified valid item-sets.
- Pattern growth approaches (e.g. Eclat [5] and FP-growth [4]) eliminates the need of explicit candidate generation with special data structures for database representation and operations.

Apriori, Eclat, FP-growth are the basis of many other algorithms.

3 BITTABLE BASED FREQUENT ITEMSET MINING

Recently Dong and Han proposed an algorithm named as BitTableFI [6]. In the algorithm, a special data structure BitTable is used horizontally and vertically to compress database for quick candidate item-sets generation and support count, respectively. But the BitTableFI suffers from the high cost of candidate generation and test.

Song et al, proposed a new algorithm Index-BitTableFI[7]. It also uses BitTable horizontally and vertically. To make use of BitTable horizontally, index array and the corresponding computing method are proposed. By computing the subsume index, those itemsets that co-occurrence with representative item can be identified quickly by using breadth-first search at one time. Then, for the resulting itemsets generated through the index array, depth-first search strategy is used to generate all other frequent itemsets. However, Index-BitTableFI always uses a fixed size of Bit-Vector for each item (equal to number of transactions in a database). It leads to consume more memory for storage Bit-Vectors and the time for computing the intersection among bit-vectors [8,9].

Janos proposed a novel algorithm [10] based on BitTable (or bitmap) representation of the data. Data - related to frequent item-sets - are stored in spare matrices. Simple matrix and vector multiplications are used to calculate the support of the potential n+ 1 item-set. Even though novel bitmap-based approach is simple but involves more matrix multiplications which lead to increase the computing.

Vo et al, proposed the dynamic bit vectors [8,9] algorithm for constructing a DBV tree and mining FIs from a database. This algorithm shows the better performance result but still it involves computation complexity by constructing DBV tree.

Saleem et.al proposed the Compact BitTable approach for mining Frequent Itemsets (CBT-fi) algorithm[11], which clusters (groups) the similar transaction into one and forms a compact bit-table structure which reduces the memory consumption as well as frequency of checking the itemsets in the redundant transaction.

4 MOBILE AGENT BASED DISTRIBUTED ASSOCIATION RULE MINING

This section presents the brief overview of mobile agents and existing agent based framework for distributed association rule mining. Software Agents refers to intelligent program that performs certain tasks on behalf of the user. Software agents endowed with the property of mobility are called Mobile Agents [MA]. MA is an autonomous transportable program that can migrate under its own or host control from one node to another in the heterogeneous network to perform a task. In other words, the program running at a host can suspend its execution at an arbitrary point, transfer itself to another host or request the host to transfer it to its next destination and resume execution from the point of suspension. Once the agent is launched, it can continue to function even if the user is disconnected from the network. MA not only moves from one host to another but also spawns new agents; interact with other stationary agents and searches services/resources [12]. Agents can support and enhance the knowledge discovery process in many ways. For instance, agents can contribute to data selection, extraction, preprocessing, and integration, and they're an excellent choice for peer-to peer parallel, distributed, or multisource mining. Agents are also a good match for interactive mining, human centered DM, service delivery, and customer service [13]. Few researchers deployed, the MA to mine the association rules from distributed sites. Some of the important contributions in this domain are presented below.

Pham Nguyen et.al[24] presented a distributed algorithm for mining association rules using the Apriori algorithm and the MA technology. to improve efficient operations while finding frequent item-sets.

Yun-Lan Wang et.al[20] proposed IDMA architecture which shows mobile agent based distributed and incremental association rule mining in distributed , heterogeneous database system. The system includes the distributed knowledge discovery management system (KDMS), the knowledge discovery sub-system (sub-KDS), the data mining mobile agent (DMMA) and the local knowledge base (LKB). The KDMS dispatches the mobile agent

DMMA to each site. The mobile agents move to the sub-KDS and execute the mission of data mining. The local large item-set scan is got so the local association rules can be obtained and the local knowledge base can be refreshed. The set of local large item-sets and their support counts led back to the KDMS by the mobile agents. When all the mobile agents come back to KDMS, the possible minimum and maximum support counts of the potential global item-sets can be got. This system was implemented based on IBM Aglet.

Cristian Aflori et.al [15] discovered association rules in a distributed database using intelligent agents and applying loose couple incremental approach for effective distributed association rule mining.

U.P kulkarni et.al.[16] suggested a method called mobile agent based distributed data mining. It basically uses 2 steps.

a. Local sites send LFIs to central site and also to all their neighbors.
b. Calculation of GFI/CGFI at central site and counts of CGFI at local sites is done as a overlapped operation. That is, local sites need not wait for central site to send CGFI. Thus total time taken is reduced drastically.

Gongzhu Hu et.al. [17]proposed an agent based approach to mine the association rules from the distributed data sets across the multiple sites while preserving the privacy of the local datasets. This approach relies on the local systems to find the frequent itemsets that are encrypted and the partial results are carried from site to site. System has one agent server which can communicate to the local hosts through six types of agents (ESUA, DSUA ,ESA, DSA, BA and OA) created and dispatched by it. The agents are defined as

a. Encrypt Secure Union Agent (ESUA)– performs encryption of each locally frequent k-itemset at host i.
b. Dencrypt Secure Union Agent (DSUA)- travels through every host to pursue decryption.
c. Encrypt Sum Agent (ESA)- travels through all the hosts to obtain the encrypted support count.
d. Decrypt Sum Agent (DSA)- carries the array of RuleSet extracted from the returned ESA. It travels through all the host and let each host subtract the random number they generated when dealing with the ESA.
e. Broadcast Agent (BA)- When the DSA agent comes back to agent master the globally frequent k-itemsets (F_k) can be calculated from the decrypted ruleset then BA is used to carry F_k to each host to update their knowledge.
f. Over Agent (OA): Its used to notify all the hosts that algorithm has terminated.

Local Hosts: there are several local hosts where agents can visits and perform their tasks of local association rule mining, encryption and decryption etc.

A. O. Ogunde [18][19] proposed adaptive architectural framework called *Adaptive Mobile Agent Association Rule Miner (AMAARM)* that mines association rules across multiple data sites, and more importantly the architecture adapts to changes in the updated database and the mining environment giving special considerations to the incremental database. This system was made adaptive both at the algorithm level and the mining agent level. Adaptation at the mobile agent level uses sensors to sense environmental changes, creates a percept of the environment and sends it to the adapter which adapts to the environmental changes by dynamically changing the goals of the mining agents or maintaining the original goals. The system promises to efficiently generate new and up-to-date rules while also adapting to faults and other unforeseen circumstances in the distributed association rules mining environment without the usual user's interference. The model presented here provided the background ideas needed for the development of adaptive distributed association rule mining agents.

Kamal Ali Albashiri et.al.[21], introduced an extensible agent enriched data mining (AEDM) system implemented in Java agent development environment (JADE) called Extendible Multi-Agent Data mining System(EMADS). The system's operation is described and illustrated in terms of two knowledge Discovery in Data (KDD) scenarios: (a). Meta Association rule mining (Meta ARM) and classifier generation. Meta mining is defined as the process of combining the individually obtained results of N applications of a data mining activity. Each local frequent item-sets T-Tree is collected and a merged T-Tree is generated for global frequent item-sets.

Kamal Ali Albashiri [22], implements data partition technique for parallel and distributed ARM. The system distributes the data among the agents in vertical/horizontal partition basis and uses Apriori-T algorithm to mining the association rules. The aim of the scenario is to demonstrate that the MADM approach is capable of exploiting the benefits of parallel computing.

Walid Adly et.al [23] presented distributed bit-table multi-agent association rule algorithm combines the association rules using the bit-table data structure and multi agent technique to decrease the time needed for candidate generation and the support count processes. BitTable data structure is very compressed and can easily fit in memory and it was implemented before the first iteration. This had a great impact on the algorithm performance.

G.S.Bhamra et.al [12] proposed framework, AeMSAR (Agent Enriched Mining of Strong Association Rules) highlights the agent based approach for mining the strong association rules from distributed data sources. This framework consist of one central site ($S_{CENTRAL}$) where global knowledge is computed and n distributed sites {S_i, i=1…n} where horizontal partitioned transaction datasets {DB_i, i=1…n} are stored. Synthetic Transactional Data sets are generated and stored at each distributed site using a tool called TDSGenerator. $S_{CENTRAL}$ acts as the agent launching station from where mobile agents are dispatched carrying some information and returned back with results. Mobile as well as Stationary agents are stored in Agent Pool at this site. A Central Security Agency (CSA) at this site assigns a legal certificate to every mobile agent before its launch and when that agent reaches at a node in its itinerary authenticity of this certificate is verified again so that no malicious agent can attack local node. There are five agents in the architecture, three of these are MAs and other two are stationary intelligent agents to perform different tasks. Mobile Agents are – Local Frequent Itemset Generator Agent (LFIGA), Local Knowledge Generator Agent (LKGA), Total Frequent Itemset Collector Agent (TFICA). These agents maintains dynamic itinerary, whenever required this can be updated at any node at any time in the itinerary. These agents maintain two containers- Result container and State container. One for transporting result data across the network and other for state variables and their intermediate values. Stationary Agents are – Global Frequent itemset Generator Agent (GFIGA) and Global Knowledge Generator Agent (GKGA).

a. ***LFIGA:*** It is launched from the $S_{CENTRAL}$ carrying given minimum threshold support (min_sup) and visits n sites in its itinerary. It generates and stores the list of local frequent k-itemsets and list of support count of every items in a site S_i by applying Apriori algorithm on the local transactional dataset (DB_i) at that site with the constraint of min_sup.
b. ***LKGA:*** It is also launched from the $S_{CENTRAL}$ and visits n sites in its itinerary. It applies the constrains of given minimum threshold confidence (min_conf) to generate and store the list of locally strong association rules by using local frequent k-itemset and list of support count generated by LFIGA at site S_i.
c. ***TFICA:*** It is also launched from the $S_{CENTRAL}$ and visits n sites in its itinerary. While visiting each site, it collects lists of local frequent k-itemset generated by LFICA and carries back the list of total frequent k-itemset at $S_{CENTRAL}$. Local frequent k-itemset can be encrypted so that privacy of the local data can be preserved.
d. ***GFIGA:*** It is a stationary agent at $S_{CENTRAL}$ mainly used for processing the total frequent k-itemset list generated by TFIGA to generate the global frequent itemset list, which is the intersection of all the local frequent k-itemset.
e. ***GKGA:*** It is also a stationary agent at $S_{CENTRAL}$ mainly used for processing the global frequent itemset list generated by GFIGA to complete the global knowledge i.e., the list of globally strong association rules.

Saleem et.al[25] presented mobile agent based association rule mining (MADARM) framework, which deploys mobile agent for mining FI in a sequential manner focus to reduces communication overhead between server and agents.This framework consist of one knowledge server (K-Server) where the global association rule is computed, some stationary agent (SA) and one mobile agent. Entire transactional database is divided into n partitions ($D=P_1, P_2….P_n$). Partitions are located in n remote sites (S_n, n=1,2…n). Each site has stationary agent, which computes the frequent itemset based on the min_sup (minimum support count). K-Server launches the one mobile agent (MA) which carries secret-key, list of sites (LS), empty GFIL(Global Frequent Itemset List) and min-sup. This MA visits the each site as per the LS. In each site, SA verifies the key and GFIL then it updates the GFIL. Once this process is completed the MA will move to the next site. Finally it brings back the GFIL to the K-Server. Based on the GFIL, Association Rule will be generated.

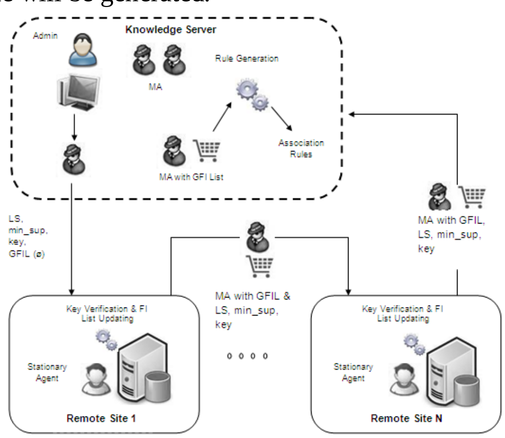

Figure 1: MAD-ARM Model

Saleem et.al[27] presented the compact bit-table (CBT-fi) based distributed association rule mining using mobile agent framework. In this framework, Entire transactional database is divided into n partitions $D = (\mathrm{D_i}, i = 1 \ldots n)$ horizontally. Partitioned Datasets are located in n remote sites$(\mathrm{S_i}, i = 1 \ldots . n)$. The framework contains KS (Knowledge Server), FICA (Frequent Item-Sets Collector Agent). Functions of each component in the framework are explained below

- KS is the knowledge server where the global frequent item-sets are generated. Using global frequent item-sets, strong association rules will be generated.
- FICA launched from KS with two containers such as LFI (local frequent item-sets) and LFISC (local frequent item-sets support count) and CBT-fi algorithm[11]. Visit each sites $(\mathrm{S_i}, i = 1 \ldots . n)$, performs LFI mining and collect the LFI and LFISC. Finally it comes back to KS.

Once the FICA comes back to KS, the compact bit table with rcv and bcv is constructed based in LFI, LFISC and thres_val. Using compact bit table with rcv, bcv GFI is generated. Finally association rules are computed using GFI.

Existing mobile agent based distributed association rule mining frameworks attempt to integrate global knowledge after the local mining and uses either existing algorithms such as Apriori , Apriori-T, FP-Tree and Bit Table for local FI mining. Especially CBT-fi based distributed association rule mining using mobile agent framework(MADARM) follow the sequential approach for mining each site using a mobile agent. This incurs more time to visit all sites. To improve the existing MADARM framework, we proposed novel approach called MADPARM, Which integrates the ideas from existing frameworks and added parallel access to each site to improves the performance of FI mining from distributed sites.

PROPOSED WORK

This section presents our proposed framework called Mobile Agent based Distributed and Parallel Association Rule Mining (MADPARM) on transactional data. This framework builds on IBM's Aglets Workbench System. The proposed framework, MADPARM is based on MADARM with parallel accessing of distributed sites with mobile agents to improve the agent communication with Knowledge Server (KS) and also improve the FI mining process by deploying CBT-*fi* algorithm in the distributed sites. We begin with problem statements.

a. *Problem Statement: Frequent Item-sets Mining*

Frequent item-sets mining is defined as follows:

Let $T = \{\mathrm{T_i} | i = 1 \ldots . n\}$ be the set of transaction in the database D and let $\mathrm{I} = \{\mathrm{I_i} | i = 1 \ldots . m\}$ be the set of items and each transaction can be identified by a distinct identifier tid.

Definition 1: A set $X \in I$ is called an itemset. An itemset with k items is called a k-itemset.

Definition 2: The support of an item-set x, denoted as $sup(x)$, is defined as the number of transactions in which x occurs as a subset.

Definition 3: For a given D, let min_sup be the threshold minimum support value specified by user. If $(x) \geq$ min_sup , item-set x is called a frequent item-set.

The task FIM is to generate all frequent item-sets in the database, which have a support greater than min_sup.

b. *Problem Statement: Distributed Frequent Item-sets Mining*

In distributed mining, global frequent item-sets are generated based on the local frequent item-sets collected from distributed sites.

S be the set of sites $S = \{\mathrm{S_i} | i = 1 \ldots . n\}$ in distributed environment.

D be the set of horizontally partitioned data sets $D = \{\mathrm{D_i} | i = 1 \ldots . n\}$ where $\mathrm{D_i}$ is the data set located in $\mathrm{S_i}$.

$$D = \bigcup_{i=1}^{n} D_i$$

KS is the knowledge server where the global frequent item-sets are generated. Using global frequent item-sets, strong association rules will be generated.

$\mathrm{LFI_i}$ - Local Frequent Item-sets of site $\mathrm{S_i}$

$\mathrm{LFISC_i}$- $\mathrm{LFI_i}$ Support Count

GFI-global Frequent Item-sets

MADPRAM also includes secured and adaptive features.

Agents used in the MADPARM are as follows:

- KS-(Knowledge Server)
- KDA-n-Key Distributor Agent

- SA-(***n***-Stationary Agent)
  - DSI-(Database Status Indicator)
  - MV- (Margin value)
  - PFI- (Previous Mined FI)
- FICA-(Frequent Item set Collect Agent)
- KS is the Knowledge Server where the Global Frequent Itemsets (GFI) are generated. Using GFI, strong association rules will be generated. KS generates ***n***-random number (p) and encrypts the p (EP) using private key and send to n-sites using KDA.
- FICA launched from KS with three containers such as LFI (Local Frequent Itemsets) , LFISC (Local Frequent Itemsets Support Count), thres_val . FICA visits each sites ($S_i, i = 1 \dots . n$) and collect the encrypted LFI and LFISC. Finally it comes back to KS.

Each local sites, stationary agent (SA) performs the following tasks:

- SA decrypts the EP using its private key to obtain the p.
- SA also checks the DSI and computes the difference (α) between DSI and current number of records (CNR) in its DB. DSI always indicates the number of records in the last FI mining. Initially DSI is zero.

  α= CNR – DSI
- SA performs FI mining using CBT-*fi* algorithm based on thres_val, only it should satisfying the following condition
  If α >= MV
  generate the LFI using CBT-*fi* algorithm. Then using p, encrypt the LFI using one-time padding algorithm and
  FICA , DSI *and* PFI will be updated.
  Otherwise FICA is updated with encrypted (One time padding algorithm) PFI using p and FICA will pass to the next site.
- Once FICA comes back to the KS, decrypt the local FI using one-time padding algorithm based on p and construct the compact bit table with rcv and bcv is constructed based in LFI, LFISC and thres_val. Using CBT with rcv, bcv , GFI is generated as shown in Knowledge Server algorithm. Finally, association rules are computed using GFI.

Knowledge Server: pseudo code

```
function Server(list_of_sites, thres_val)
begin
MA=∅;
vector GFI=∅;
vector LFI=∅;
vector LFISC=∅
p=random(255);
EP=Encrypt(p, private_key);
Launch KDA(EP);
if (visted_sites<>∅)
        MA =launch FICA(list_of_sites, thres_val, LFI, LFISC);
while(MA.count <> 0)
        Begin
            Decrypt(LFI,LFISC,p);
            GFI=CBTFI(LFI,LFISC, thres_val) // global frequent item generation
        end
end
```

SA: pseudo code

```
Begin
        Scan database D_i once and count the number of records (CNR)
            α= CNR – DSI
            If(α >= MV) then
                Construct compact bit table using thres_val
                Compute FI with support count (SC)
                Encrypt(LFI,SC,p)
                Update PFI
                Update FICA(FI,SC, SID)
                DSI = CNR
            Else
                Update FICA(PFI,SC, SID)
End
```

Cost of communication (CC) is based on the total time taken to compute the GFI can be stated as follows

$T_{IS} = Time\ taken\ to\ initialize\ server$

$TLFICA_i = Time\ to\ launch\ FICA\ to\ site\ i$

$TRFICA_i = Time\ to\ return\ FICA\ from\ site\ i$

$$TTRFICA = \sum_{i=1}^{n} TRFICA_i$$

$- TTRFICA\ (Total\ Turn\ around\ time\ of\ all\ FICA)$

$T_{GFI} = Time\ taken\ to\ compute\ GFI\ at\ server$

$CC = T_{IS} + TTRFICA + T_{GFI}$

5 EXPERIMENTAL RESULTS

Experiments were conducted to show the performance of the two framework (MADARM and MADPARM) in terms of time taken to computing GFI. There are two synthetic datasets are used for analysis which are based on retail database from FIMI repository. The characteristics of two datasets that we used in the experiments are shown in table 1. Figure 2 shows the mining time of these dataset by varying threshold. The performance of CBT-*fi* algorithm based MADARM and MADPARM is compared. The results show that MADPARM performs better than the CBT-*fi* algorithm based MADARM.

Table 1: Synthetic datasets based retail database

Name	D	T	N
T30D300N70	300	30	70
D (Total number of transaction)			
T(Average number of items in a transaction)			
N (Total number of items)			

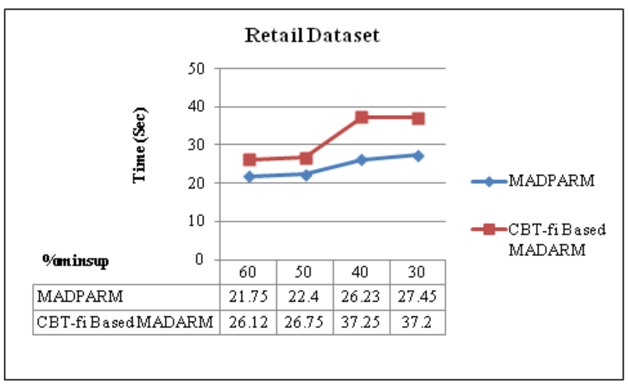

Figure 1: T30D300N70 with 3 sites

REFERENCES

[1]. L.Cao(ed.), Data Mining and Multi-agent Integration, DOI:10.1007/978-14419-0522-2_3, pp 47-58, Springer Science + Business Media, LLC 2009.

[2]. R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Chile, 1994, pp. 487–499.

[3]. Benjamin, Rainer, Wolfgang, Memory-Efficient Frequent-Itemset Mining, EDBT 2011 ACM, Uppsala, Sweden.

[4]. Christian Borgelt. "An Implementation of the FP-growth Algorithm".In International Workshop on Open Source Data Mining, 2005.

[5]. Christian Borgelt. "Efficient Implementations of Apriori and Eclat".In Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, 2003.

[6]. Dong J, Han M., "BitTableFI: An efficient mining frequent itemsets algorithm," Knowledge-Based Systems, vol.20 no.4, pp.329-335, 2007.

[7]. Song W, Yang B, Xu Z., "Index-BitTableFI: An improved algorithm for mining frequent itemsets," Knowledge-Based Systems, vol.21, pp. 507–513. 2008.

[8]. Vo B, Hong T.P, Le B, "Dynamic Bit Vectors: An efficient approach for mining frequent itemsets," Scientific Research and Essays, vol.6, pp.5358-5368, 2011.

[9]. Vo B, Hong T.P, Le B, "DBV-Miner: A Dynamic Bit Vector Approach for fast mining frequent closed itemsets," Expert system with Applications, vol. 39, pp. 7196-7206, 2012.

[10]. Abonyi J., "A Novel Bitmap-Based Algorithm for Frequent Itemsets Mining," Computational Intelligence in Engineering Studies in Computational Intelligence, vol. 313, pp. 171-180, 2010.

[11]. Saleem Raja.A ,George Dharma Prakash Raj, CBT-fi: Compact BitTable Approach for Mining Frequent Itemsets, ACSIJ Advances in Computer Science: an International Journal, Vol. 3, pp.72-76, Issue 5, No.11 , Sep 2014.

[12]. G.S.Bhamra, A.K.Verma and R.B.Patel,"Agent Enriched Distributed Association Rule Mining: A Review". Springer Verlag Berlin Heidelberg, 2012.

[13]. Longbing Cao, Vladimir Gorodetsky, Pericles A. Mitkas, Agent Mining: The Synergy of Agents and Data Mining, IEEE Intelligent System,pp 64-72, 2009

[14]. Yun-Lan Wang, Zeng-Zhi Li and Hai-Ping Zhu, "Mobile Agent Based Distributed and Incremental Techniques for Association Rules". In Proceeding of the Second International Conference on Machine Learning and Cybernetics", 2003.

[15]. Aflori, C., Leon, F.: Efficient Distributed Data Mining using Intelligent Agents, Proceedings of the 8th International Symposium on Automatic Control and Computer Science (IASI), ISBN 973-621-086-3 (2004)

[16]. U.P.Kulkarni, P.D.Desai ,Tanveer Ahmed , J.V.Vadavi, A.R.Yardi, Mobile Agent Based Distributed Data Mining, International Conference on Computational Intelligence and Multimedia Applications , IEEE Computer Society 2007.

[17]. Gongzhu Hu and Shaozhen Ding, An Agent-Based Framework for Association Rules Mining of Distributed Data, Software Engineering Research, Management and Applications, SCI, vol- 253, pp 13-26, Springer-Verlag Heidelberg 2009.

[18]. Gongzhu Hu and Shaozhen Ding , Mining of Assocation Rules from Distributed Data using Mobile Agents, ICE-B 2009, PP 21-26 (2009).

[19]. A. O. Ogunde, Olusegun Folorunso, Adesina Simon Sodiya, On the Adaptivity of Distributed Association Rule Mining Agents, The Fourth International Conference on Adaptive and Self-Adaptive Systems and Applications, 2012.

[20]. A. O. Ogunde, O. Folorunso, A. S. Sodiya, and J. A. Oguntuase, "Towards an adaptive multi-agent architecture for association rule mining in distributed databases," Adaptive Science and Technology (ICAST), 2011 3rd IEEE International Conference on 24-26 Nov. 2011, pp. 31 – 36 from IEEE Xplore.

[21]. Albashiri, K.A., Coenen, F.: Agent-Enriched Data Mining Using an Extendable Framework, ADMI 2009, LNCS 5680, PP 53-68 , Springer Verlag Berlin Heidelberg (2009).

[22]. Albashiri, K.A.: Agent Based Data Distribution for Parallel Association Rule Mining, International journal of computers, vol 8, 2014.

[23]. Walid Adly Atteya, Keshav Dahal and M.Alamgir Hossain, "Distributed BitTable multi-agent Association Rules Mining Algorithm" , Springer-Verlag, KES 2011, Part I, LNAI 6881.
[24]. Pham Nguyen Anh Huy, Ho Tu Bao, A distributed algorithm for mining association rules, Proceedings of The Third International Conference on Parallel and Distributed Computing, Applications and Technologies 2002.
[25]. Saleem, George, "MAD-ARM: Mobile Agent based Distributed Association Rule Mining", ICCCI'13, IEEE Conference 2013.
[26]. Saleem, George, Mobile Agent based Distributed Association Rule Mining: A Survey, 2nd International Conference on Applied Information and Communications Technology(ICAICT) –Elsevier – 2014.
[27]. Saleem, George, Compact BitTable based Distributed Association Rule Mining using Mobile Agent Framework, European Journal of Scientific Research, Volume 128 No 4, January, 2015.