# Improved cost models for agent-based association rule mining in distributed databases

**Ogunde A. O.**
**Computer Science Programme, Department of Mathematical Sciences, Redeemer's University (RUN), Redemption City, Ogun State, Nigeria**

**Folorunso O., Sodiya A. S., Oguntuase J. A.**
**Department of Computer Science, University of Agriculture, Abeokuta, Ogun State, Nigeria**

**Ogunleye G. O.**
**Computer Science Programme, Department of Mathematical Sciences, Redeemer's University (RUN), Redemption City, Ogun State, Nigeria**

**ABSTRACT.** Distributed systems have really gained ground in several applications domains. Recently, extracting valuable knowledge from large volumes of data in distributed setting is a major concern for most organizations. There is therefore need for architectures and models for efficient mining associations and connections from such data. Many models have been produced but no one model fits all scenarios. Massive data movement, high communication costs and ultimately slow response time are some major problems with existing models. This work therefore proposes an improved cost model for mining association rules from distributed databases.
**KEYWORDS:** Data mining, Association rule mining, Knowledge Integration, Cost Models, Mobile Agents, Response Time.



## Introduction

Advances in hardware development, especially the development of large space storage, has really enabled lots of organization build large storage

database and collect large volume of data. Those organizations have the desire to extract useful information from the ultra large amount of data. So some traditional way will be not enough to handle the information. Association rule mining (ARM), fist proposed by [AIS93], tries to find frequent patterns, associations, correlations, or casual structures sets of items or objects in transaction database, relational database, etc. The idea is to find out the relation or dependency of occurrence of one item based on occurrence of other items. Lots of algorithms in this area had been proposed.

Data mining is known to be among the recent topics in computer science. According to [Pau10], many current data mining tasks can only be accomplished successfully in a distributed setting. Therefore, new methods for mining vast amounts of heterogeneous data from several data sources are emerging all the time [BF07]. Distributed association rule mining is currently not very actively researched and there are only few models available for estimating the response time of these systems. Recent papers are focusing on minimizing the communication cost of the prior algorithms and are taking dynamic datasets into account.

Software agents have really evolved in distributed computation paradigm. Mobile agent is a thread of control that can trigger the transfer of arbitrary code to a remote computer. Mobile agents' paradigm has several advantages: Conserving bandwidth and reducing latencies. Also, complex, efficient and robust behaviors can be realized with surprisingly little code. Mobile agents can be used to support weak clients, allow robust remote interaction, and provide scalability.

In this research, problems related to Association Rule Mining in distributed data will be examined. An optimized model for estimating the response time of Distributed Association Rule Mining (DARM) is presented. This research will mainly focus on distributed association rule mining algorithms as the building block of the approximate distributed algorithms. Existing systems will be analyzed and a novel architecture will be presented as original contributions of this research work.

## 1. Literature Review

Review of related works is carried out in this section.

### 1.1. Data Mining (DM)

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data set. It is a powerful new technology with great potential to help companies focus on the most important information in the data they have collected about the behavior of their customers and potential customers [RV10]. Consequently, data mining has become an important research area [CHY96]. Data mining differs from other data analysis techniques in that the system takes the initiative to generate patterns by itself.

### 1.2. Distributed Data Mining (DDM)

When data mining is undertaken in an environment where users, data, hardware and the mining software are geographically dispersed, it is called distributed data mining. DDM therefore refers to the mining of distributed data sets. The data sets are stored in local databases hosted by local computers which are connected through a computer network [ATS04]. Typically, such environments are also characterised by heterogeneity of data, multiple users and large data volumes [CK97].

DDM offers techniques to discover knowledge in distributed data through distributed data analysis using minimal communication of data [KIB00]. Typical DDM algorithms involve local data analysis from which a global knowledge can be extracted using knowledge integration techniques [KIB00]. A typical DDM framework is shown in figure 1.

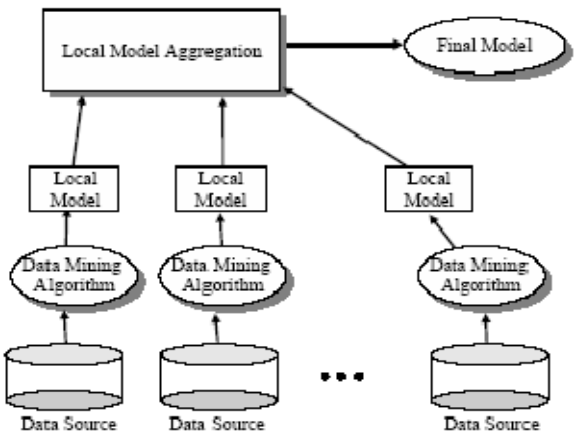


**Figure 1: A Typical DDM Framework (Source: [RV10])**

### 1.3. Association Rule Mining (ARM)

Association rule mining is the discovery of associations or connections among objects. Since its inception, association rule mining has become one of the core data-mining tasks and has attracted tremendous interest among researchers and practitioners. An association rule is in the form of "A1 ^ …… ^ Ai → B1 ^ …….. ^ Bj" which means objects B1; …… ;Bj tend to appear with objects A1; ……..;Ai in the target data. Association rules at multiple conceptual levels will reveal such kind of association in the relevant set(s) of data in a database.

Algorithms for association rules discovery are in two steps. All frequent itemsets are found in the first step. The frequent itemset is the itemset that is included in at least *minsup* transactions. The association rules with the confidence at least *minconf* are generated in the second step. The formal definitions of these two metrics are:

$$Support, s(X \Rightarrow Y) = \frac{\sum(X \cup Y)}{N}$$

$$Confidence, c(X \Rightarrow Y) = \frac{\sum(X \cup Y)}{\sum X}$$

According to [HGN00], there is a border that separates the frequent itemsets from the infrequent ones - thus; the problem is restricted on finding that border. The formal definition of association rule mining is: Let I = $\{i_1,i_2,\ldots,i_m\}$ be a set of literals called items and D be a set of transactions where each transaction T is a set of items such that $T \subseteq I$. Associated with each transaction is a unique identifier, called its TID. We say that a transaction T contains X, a set of some items in I, if $X \subseteq T$.

### 1.4. Existing DDM Systems

Several DDM systems have been proposed in the literature. A comprehensive review of issues and challenges in current agent-based DARM has been done by [O+11]. In [KHS97, PADMA system was presented. The PADMA system employed the approach of a central-learning strategy. They described a parallel DM system (PADMA) that uses software agents for local data accessing and analysis, and a web based interface for interactive data visualization. Partial data cluster models are first computed by stationary agents locally at different sites. All local models are collected to a central site that performs a second-level

clustering algorithm to generate the global cluster model. PADMA has been used in medical applications.

In [S+97], Java Agents for Meta-learning (JAM) over distributed databases was proposed. JAM provided a set of learning programs implemented either as JAVA applets or applications that computed models over data stored locally at a site. JAM supported the launching of learning and meta-learning agents to distributed database sites. Papyrus [B+99] is a Java-based system addressing wide-area DDM over clusters of heterogeneous data sites and metaclusters. It supports different task and predictive model strategies. Mobile DM agents move data, intermediate results, and models between clusters to perform all computation locally and reduce network load, or from local sites to a central root which produces the final result. Each cluster has one distinguished node which acts as its cluster access and control point for the agents. Coordination of the overall clustering task is either done by a central root site or distributed to the (peer-to-peer) network of cluster access points. Papyrus supports various methods for combining and exchanging the locally mined predictive models and metadata required to describe them by using a special markup language.

Theoretical models for Client/Server, mobile agents and hybrid DDM models were proposed in [KLZ00]. They proposed theories that combined client-server and distributed models. Not much work was done in the paper as the researchers concluded that the work still needs improvement and experimental validation. [AMM03] also proposed an OIKI model: e-business model for DDM using mobile agents. OIKI model is a mobile agent based DDM model that overcome the drawbacks of the traditional mobile agent based DDM model. Instead of transferring the results from each data server to the client, the client controls migration of the results among data servers to be integrated locally and finally, the final results are transferred to the client. Only few conceptual views are presented in the paper while the researchers concluded that the implementation issues of their proposed model still have a lot of work to be done.

[Sil06] studied Distributed and Stream Data Mining Algorithms for Frequent Pattern Discovery in his PhD thesis. They offered a perspective on DDM algorithms in the context of multiagent systems, discussing broadly the connection between DDM and MAS and also providing a high-level survey of DDM with the main focus of their work on distributed clustering algorithms and some potential applications in multi-agent-based problem solving scenarios. [BF07] gave the state of DDM in their paper focusing mainly on the state of the art in distributed clustering and frequent itemset mining.

[ACL09] worked on EMADS, an Extendible Multi-Agent Data mining System. The EMADS vision is that of a community of data mining agents, contributed by many individuals, interacting under decentralised control to address data mining requests. EMADS is seen both as an end user application and a research tool. The paper detailed the EMADS vision, the associated conceptual framework while concentrating mainly on agent based data classification. [RV10] in their work combined data mining wit game theory as a way of striking a balance and reaching a steady state between the data miner and the adversary in adversary data mining.

[PS08] proposed a knowledge integration method in a parallel and distributed environment with association rule mining using XML data. A scanty write-up on the knowledge integration aspect was done and how the real knowledge integration of the XML data would be done was not presented in the paper. [Pau10] also proposed an optimized algorithm for distributed and parallel data mining but with XML data, because of the complexity of XML data. In their approach, multiple nesting problems in XML data was handled appropriately to assure the correctness of their result.

Common to all approaches is that they aim at integrating the knowledge which is discovered out of data at different geographically distributed network sites with a minimum amount of network communication, and maximum of local computation. All these studies have presented a different data mining scenarios involving various architectural models, which are still very open for further research. EMADS [ACL09] is the closest to this work but it uses a distributed algorithm and is mainly based on agent-based data classification while the architecture proposed in this system will be purely based on agent-based association rule mining. The drawback of the system is that after mining, all the individual data results have to migrate to back to the requesting server which may incur serious communication costs and also bringing large set of results together on one the requesting system may pose serious challenges in term of memory constraints. Also, since the algorithm is distributed, any fault at any of the data sites may hinder the successful completion of the mining process.

### 1.5. Agents and Multi-Agent Systems

Agents are defined by [Woo09] as computer software that are situated in some environment and are capable of autonomous action in this environment in order to meet their design objectives. Intelligent agents [Rud04; Woo09] are defined as agents that can react to changes in their environment, have social ability (communication) and the ability to use

computational intelligence to reach their goals by being proactive. Agents are active, task-oriented, modeled to perform specific tasks and capable of autonomous action and decision making.

By combining multiple agents, in one system, to solve a problem, the resultant system is a Multi-Agent System (MAS) [Woo09]. These systems are comprised of agents that individually solve problems that are simpler than the overall problem. They can communicate with each other and assist each other in achieving larger and more complex goals. MAS may consist of possibly heterogeneous agents that are scattered around the environment and act autonomously or in collaboration. Well documented advantages of MAS according to [Woo09] include: Decentralised control, Robustness, Simple extendability, Sharing of expertise and Sharing of resources.

### 1.6. Mobile Agent Based DDM Model

The agent-based model is a popular approach to constructing distributed data mining systems and is characterized by a variety of agents coordinating and communicating with each other to perform the various tasks of the data mining process. Agent technology is seen as being able to address the specific concern of increasing scalability and enhancing performance by moving code instead of data and thereby reducing the communication overhead incurred in the Client-Server (CS) model.

Mobile agents do not waste the bandwidth, because the agent migrates to the server. The agent performs the necessary sequence of operations locally, and returns just the final result to the client, [G+00]. The major drawback in the CS-based DDM model is that huge amount of data sets migrate form the data sources locations to the DM sever to accomplish the required DM process. This results into a considerable waste in the network bandwidth and consequently a big increase in latency. A typical mobile agent-based DDM process begins with a client request for a DM process. The client determines the required data severs for the DM process and multicasts a set of mobile agents data miners MADMs. The MADMs migrate to the data servers and perform the data mining operations locally and return the final results (knowledge) to the client. Finally, the client uses a knowledge integration (KI) program to integrate the DM results from the different MADMs.

## 2. The DARM Architecture

The DARM architecture proposed here utilizes the mobile agent paradigm to build Multi-Agent Distributed Association Rule Miner (MADARM) for distributed databases. Intelligent agents' paradigm used here will help our architecture to overcome the communication bottleneck associated with huge size of data migrated in traditional DARM systems.

The proposed architecture is characterized by a given distributed data mining task being executed in its entirety using the mobile agent paradigm. In general, this can be expressed as **m** mobile agents traversing **n** data sources. Here **m = n+1** is deployed**,** which is the case where one mobile agent called MAARM (Mobile Agent-Based Association Rule Miner) is deployed to each data source. The Association Rule Mining Coordinating Agent (**ARMCA**) is the main coordinating agent resident in the agent zone of the DARM server. The ARMCA creates the MAARMs, one for each data source. The ARMCA later creates an additional mobile-agent called the Results Integration Coordinating Agent (**RICA**) for optimization of the knowledge or results integration between all the data sources. The algorithm below shows the core steps that will be involved.

*Step 1: submission of a task by a user to the DARM server through the user interface*
*Step 2: mining request sent to the agent zone where ARMCA coordinates the entire agent-based mining process*
*Step 3: dispatching of mobile agents – MAARMs with the appropriate DM algorithm to the respective data sources by the coordinating agent*
*Step 4: distributed association rule mining at each data source by the ARM agents - MAARMs*
*Step 5: return of information concerning results sizes at each data site by the Mobile Agent-Based Results Reporters (MARRs)*
*Step 6: coordination of appropriate and effective knowledge integration based on sizes of results by the coordinating agent – ARMCA at the agent zone*
*Step 7: dispatching of a mobile agent – RICA with the appropriate KI algorithm to the respective result sites by the coordinating agent for knowledge integration*
*Step 8: integration of results within the result sites my RICA using the model presented in this research*

*Step9: the return of integrated results (association rules) by RICA to the agent zone before presentation to the user through the user interface.*

The system described here is depicted by the architecture in figure 2.

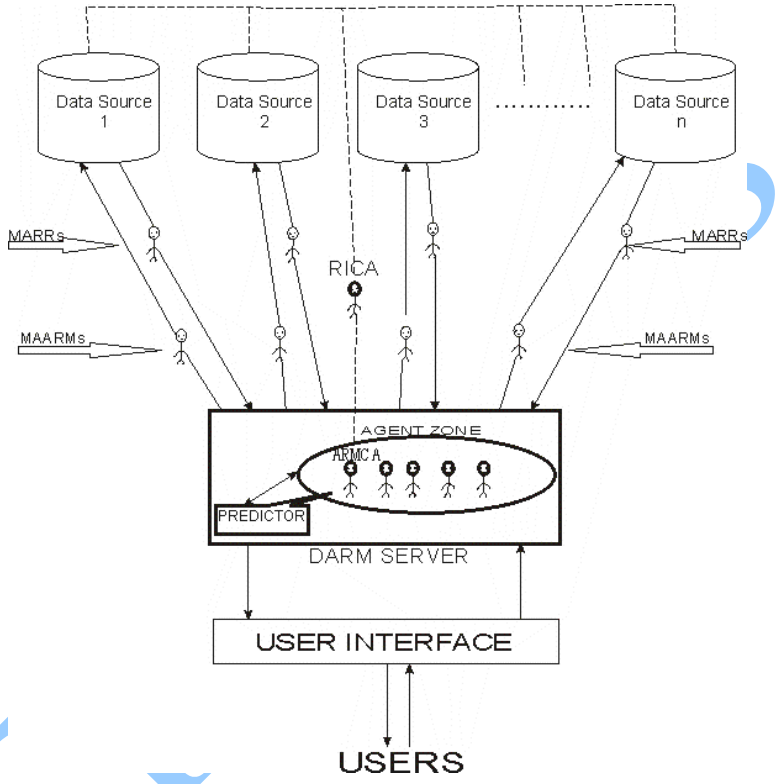


**Figure 2: The Proposed DARM Architecture**

This is elaborated in these five main stages.

**Request Stage**: Request for association rule mining is initiated at the DARM Server by the user.

**Preparation Stage**: The DARM server through the ARMCA (resident in the agent zone) multicasts MAARMs (Mobile Agent-Based Association Rule Miners) equipped with either the *Apriori* or the *FP-Growth* algorithms depending on the size of the data at each data source. All activities of the MAARMs including allocation of suitable mining algorithm to each agent will be fully coordinated by the ARMCA.

**Association Rule Mining Stage**: Association rule mining process is performed locally by the MAARMs on each data source. After which the result is kept by the MAARM agent while only the result information is passed to the MARRs (Mobile Agent-Based Result Reporter), which are new agents created by each MAARM, so as to reduce the size of agents and massive data movement in the system. The MARRs report only results information after the mining process to the ARMCA. After this the MARRs are killed by the ARMCA.
**Knowledge Integration Stage**: Knowledge integration is optimized on the data sources by using a new approach - optimized agent-based distributed knowledge integration (ADKI) as opposed to Incremental Knowledge Integration proposed by [AMM03] which does not take into consideration the size of the agent, size of the results, and bandwidth. Formal methods for achieving these are major considerations in this research. Here, the MARRs report the results to the ARMCA at the agent zone. An agent-based approach will be used to model the distributed knowledge integration at this stage through the *predictor* at the DARM server. ARMCA will use the *predictor* to model which result *j* is integrated with result *k* at which data source *n*. The *predictor's* decision will be based on the time costs of agent transfer, knowledge transfer and results integration. These knowledge or results integration will be carried out by the Results Integration Coordinating Agent (RICA), which is a new single mobile agent created by ARMCA for the purpose of performing the optimized results integration. RICA knows exactly which result *j* to integrate with result *k* first, second and so on. RICA integrates the results of two data sources at a time. This process is repeated until all integrated results are resident in a specific data source.
**Results Migration Stage**: only the overall integrated mining results are migrated back to the DARM requesting server through the RICA.

## 2.1. Algorithm Consideration

The proposed model will be based on the state-of-the-art Apriori algorithm [AS94] called *Apriori*, which is a breadth-first search algorithm with candidate generation, to test for the comparatively sparse databases. Likewise, the state-of-the-art depth-first search partitioning algorithm pattern growth method, *FP-Growth* algorithm [HPY00] will also be used for comparatively dense databases. Estimates of the distributed association rule mining response time for a given task would be carried out by the predictor in our design. The choice of algorithm to be embedded into mining agent -

MAARM by the coordinating agent - ARMCA would be determined by the size of the dataset at each data site. For the purpose of this work, the entire dataset would fall into two categories: either sparse dataset (data size interval 1) or dense dataset (data size interval 2).

### 2.2. Optimized Cost Models Associated With Distributed Association Rule Mining DARM)

The proposed architecture will perform the distributed association rule part using three estimates:

• Agent cost estimates, which involve the time needed for the Mobile Agent-Based Association Rule Miners (MAARMs) to travel from the agent zone to the data sources

• Local Association Rule Mining costs, which are estimate of the time needed for association rule mining locally at each data source.

• Results Information Transfer costs, which are estimates of the time needed for MARRs to travel back to the agent zone with results information concerning each local mining.

#### 2.2.1. Optimized Cost Models Associated With Distributed Knowledge Integration (DKI)

The proposed architecture will perform the optimized knowledge integration strategy for a distributed association rule mining task using three estimates:

• Agent cost estimates, which involve the time needed for the knowledge and results integration agent RICA to travel from the agent zone to the first data server

• Knowledge Integration costs which are estimate of the time needed for knowledge integration between the data servers.

• Results transfer costs which are estimate of the time needed for RICA to travel back to the agent zone with only final integrated results.

Based on these estimates, the optimizer chooses the best options for the knowledge integration process. Therefore, the optimizer's tasks involve building cost estimates and then choosing the best alternative. In order to further optimize the system, the Distributed Knowledge Integration Simulator (DKIS) in the architecture will maintain a database of information about previous knowledge integration tasks and their response's times. The simulator will use this stored information to later predict time costs estimates for future distributed knowledge integration involving similar cases in similar environments. This will also save more time in the system

as the optimizer will not need to calculate the estimated response time for the case at hand but uses already stored estimates from the simulator provided the parameters are similar.

The overall response time for the distributed association rule mining would be calculated as follows:

$$T = t_{darm} + t_{dki} \quad \text{..... (1)}$$

where T is the response time, $t_{darm}$ is the time taken to perform mining in a distributed environment and $t_{dki}$ is the time taken to perform distributed knowledge integration and return the results to the requesting server.

### 2.2.3. Optimized Cost Models for the proposed Architecture

Let n be the number of data servers. Let the number of mobile agents deployed be m. In order to derive the time function for the general case involving n data servers and m data mining agents, we first formulate the cost function for the case where there is one data server and one data mining agent.

### 2.2.4. Cost Model for One Data Server and One Mining Agent

Let us consider the case where data mining has to be performed at the ith data server (i.e $1 \leq i \leq n$ ). Let $t_{mobileAgent}(x, y)$ refers to the time taken by the mobile agent to travel from node x to node y.

The time function for the response time to perform distributed association rule mining involving *i* data servers is as follows:

$$t_{darm} = t_{MAARM}(AZ, i) + t_{arm}(i) + t_{MARR}(i,AZ) \quad \text{..... (2)}$$

where,
$t_{darm}$= time taken for distributed association rule mining
$t_{arm}(i)$ = time taken for association rule mining at data server i
$t_{MAARM}(AZ,i)$ = time taken for MAARM (embedded with the mining algorithm and other parameters) to move from the Agent Zone to the *ith* data server
$t_{MARR}(i,AZ)$ = time taken for MARR to move from ith data server back to the Agent Zone

In this case $t_{dki} = 0$ (this is because there is no need for knowledge integration for this particular case).

This implies that (from (1)), total response time for DARM is

$T = t_{MAARM}(AZ, i) + t_{arm}(i) + t_{MARR}(i,AZ)$ ..... (3)

In general, the time taken for a mobile agent to travel depends on the following factors: the size of the agent and the bandwidth between nodes (e.g. in kilobits per second). The travel time is proportional to the size of the agent and is inversely proportional to the bandwidth (i.e. the time taken increases as the agent size increases, and decreases as the bandwidth increases). This can be expressed as follows:

$T_{MAARM}(AZ, i) \propto$ size of darmAgent ..... (4)
$T_{MAARM}(AZ, i) \propto 1$ / bandwidth ..... (5)

From (4) and (5):

$T_{MAARM}(AZ, i)$=(k*size of MAARM) /(bandwidth between AZ and i)..(6)

In the above expression for the time taken by the data mining agent to travel from the agent zone to the data server, k is a constant.

In Stolfo et. al. (1997], the size of an agent is given by the following triple, size of an agent = <Agent State, Agent Code, Agent Data> where, agent state is the execution state of the agent, agent code is the program that is encapsulated within the agent that performs the agent's functionality and agent data is the data that the agent carries (either as a result of some computation performed at a remote location or the additional parameters that the agent code requires). Applying this to our work gives,

*size of a MAARM=<MAARM state, DARM algorithm, input parameters>*(7)

$t_{MAARR}(i, AZ)$ is the time taken for the data mining results to be transferred from the data server (i) to the agent zone (AZ) is estimated similarly to the MAARM.

This agent does not carry code to be executed, but merely transfers the results information to the agent zone for knowledge integration. It must be noted that unlike the time taken by the data mining agent to travel from the agent zone to the data source, the time taken for the result to be carried cannot be estimated a priori since size of the results depends on the characteristics of the data.

### 2.2.5. Optimized Cost Model for General Case DARM involving *m* Mining Agents and *n* Data Servers

We now extend the cost estimate for the general case characterised by n mobile agents and n distributed data sources. Let there be n data sources which need to be accessed for a particular distributed data mining exercise. The agent zone dispatches n mobile agents (MAARMs) encapsulating the respective mining algorithms and parameters (i.e. one to each of the data sources) concurrently. Mining is performed at each of the sites in parallel and only the results information (not the actual results) are returned to the agent zone through the MARRs. Since the mining is performed at the distributed locations concurrently, the total time taken is equal to the time interval required by the server which takes the longest time to perform mining and return results. Therefore,

$$t_{darm} = t_{MAARM}(AZ, i) + \max(t_{arm}(i)) + t_{MARR}(i,AZ) \quad \text{..... (8)}$$

where i = 1……..n. In equation 6, the first term is the time taken by the MAARM to travel from the agent zone to data source *i*. The second term is the maximum of the times taken by the MAARMs to mine at all data source sources *i*. The third term in the expression is the time taken for the agent MARR to travel from the data source back to the agent zone with only the results information.

### 2.2.6. Cost Model for General Case DKI involving *1* RICA and *d* Initial Result Sites

We now extend the time estimate for DARM to find the time model for Distributed Knowledge Integration (DKI). The knowledge integration can only take place after all the results information are returned to the agent zone through the MARRs created by the MAARMs. The results are sorted out according to the size of each result (*rs*) from each data site in descending order by the ARMCA at the agent zone. This is because the size of the integrating agent would be less at initial stage and would be able to integrate two large-sized results faster. Towards the end of the knowledge integration, the size of the RICA would have increased and would only need to integrate the remaining small-sized results. Each result site is henceforth referenced as *r*. Note that the number of results sites still remains as the initial number of data sources, which is *n*. In this scenario, *r* ranges from 1 ……n

with $r_1$ as the largest and $r_n$ as the least (that is in descending order). The time model is computed as follows.

Let there be *n* result sites which needs to be accessed for a particular distributed knowledge integration exercise. The agent zone dispatches only *1* mobile agent called Results Integration Coordinating Agent (RICA) encapsulating the respective integration algorithm and parameters. The RICA agent is dispatched from the agent zone to the first of the result sites $r_1$. The agent carries the result of the first ($r_1$) to the second result site $r_2$ for the first knowledge integration. From there on, the agent completes the knowledge integration task and carries the results obtained to the next result site and so on, until the entire results in each of the n result sites have been integrated. The RICA then finally returns to the agent zone with the final results (association rules) prior to returning the consolidated results to the user. Therefore, the response time for distributed knowledge integration is stated in this scenario as $t_{dki}$ and is expressed in equation (7) as follows:

$$t_{dki} = t_{ARMCA}(sort(rs)) + t_{RICA}(AZ, r_1) + \sum_{r=1}^{n-1} t_{RICA}(r, r+1) + \sum_{r=2}^{n} t_{ki}(r) + t_{RICA}(r_n, AZ) \quad (9)$$

In equation 9, the term $t_{ARMCA}$ (sort(rs)) is the time taken for the ARMCA to sort the sizes of results' information (rs) at the agent zone. The term $t_{RICA}$ (AZ, $r_1$) is the time taken for the knowledge integration agent RICA to travel from the agent zone to the first result site (the result sites are already sorted from 1 through n according to their sizes in descending order). The third term is the total time taken by the knowledge integration agent RICA to travel from result site *r* to result site *r+1* as r ranges from 1 to n-1. The fourth term is the total time taken by the knowledge integration agent RICA to perform knowledge integration at each of the result sites *r* ranging from 2 to n. The estimation for time taken to perform knowledge integration at a given site and the time taken to for the agent to travel between any two sites is performed as discussed previously. However, in this case, the size of the RICA is specified as follows:

*size of RICA = < RICA state, knowledge integration algorithm, input parameters + knowledge integration results>* ..... (10)

Thus the data carried by the knowledge integration agent in this case includes both the input parameters and the knowledge integration results. This implies that the size of the agent increases incrementally as the result sites that have been integrated increases along its route.

### 2.2.7. Overall Time Model for the Proposed Architecture

The overall response time *T* for the adaptive multi-agent architecture for association rule mining in distributed databases from equation 1 is: $T = t_{darm} + t_{dki}$ where T is the response time, $t_{darm}$ is the time taken to perform mining in a distributed environment and $t_{dki}$ is the time taken to perform distributed knowledge integration and return the results to the requesting server.

Therefore, combining equations 8 and 9 we have

$$T = t_{MAARM}(AZ,i) + \max(t_{ARM}(i)) + t_{MARR}(i,AZ) + t_{ARMCA}(sort(rs)) + t_{RICA}(AZ,r_1) + \sum_{r=1}^{n-1} t_{RICA}(r,r+1) + \sum_{r=2}^{n} t_{ki}(r) + t_{RICA}(r_{n,}AZ) \quad \text{..... (11)}$$

### Concluding Remarks and Future Work

In this work, an optimized architecture model for mining association rules in distributed databases using mobile agents' scenarios involving two state-of-the-art algorithms is proposed. The work also defined optimized cost models for determination of mining results' response time at the data sources. This will greatly save data communication costs by drastically reducing movement of massive data in the system. There is no need for the usual massive data movement in global knowledge integration. Here, only the final results are migrated to the DARM server.

The proposed system is an ongoing work that is still subject to experimental validation. The system will be implemented with Java and all agents will be created in Java Agent Development Environment (JADE). Agent characteristics, agent functionalities, protocols, communication, coordination and co-operation will all be done under JADE, which is a standard agent based development platform that is robust, scalable and secure. Depending on the algorithm used for the distributed association rule mining (i.e. Apriori or FP-Growth) and the different scenarios within each model, the factors which determine $t_{darm}$ will change resulting in a consequent change in the actual time function that determines $t_{darm}$.

Finally, the cost models presented here forms a theoretical basis for the improved DARM architecture in this work and it serves a basic model that can be used by researchers in this area to estimate and predict the response time of DARM tasks. Our architecture introduced an improved

approach to ARM in distributed environments. It also indicated an improved approach to distributed knowledge integration part of DARM tasks. The architecture can intelligently switch between one algorithm or the other depending on the size of the data size and other factors related to the DARM environment. The system has a potential of reducing DARM communication/interpretation costs and response time, improve autonomy, efficiency and scalability of current DARM tasks. Implementation of this work given major considerations to the adaptivity of the DARM system are future works currently considered.

## References

[AIS93] **R. Agrawal, T. Imielinski, A. Swami** - *Mining Association Rules between Sets of Items in Large Databases*. In Proceedings of the ACM SIGMOD Conference on Management of Data, pages 207-216, 1993.

[AS94] **R. Agrawal, R. Srikant** - *Fast algorithm for mining association rules*. In Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, pages (487-499), 1994.

[ACL09] **K. A. Albashiri, F. Coenen, P. Leng** - *EMADS: An Extendible Multi-Agent Data Miner*. Volume 22, Issue 7, pages (523-528). Knowledge-Based Systems (KBS) Journal, 2009.

[AMM03] **E. I. Ariwa, B. S. Mohamed, M. M. Mohamed** - *Informatization and E-Business Model Application for Distributed Data Mining Using Mobile Agents*. International Conference WWW/Internet 2003, 2003.

[ATS04] **M. Z. Ashrafi, D. Taniar, K. Smith** – *Monash University* "ODAM. An Optimized Distributed Association Rule Mining Algorithm", IEEE distributed systems online 1541-4922 © 2004 published by the IEEE computer society vol. 5, no. 3; march, 2004

[B+99] **S. Bailey, R. Grossman, H. Sivakumar, A. Turinsky** - *Papyrus: a system for data mining over local and wide area*

*clusters and super-clusters*. In Proc. Conference on Supercomputing, page 63. ACM Press, 1999.

[BF07] **M. Byrd, C. Franke** - *The State of Distributed Data Mining*. [ECS265 Project Report] UC Davis, Davis CA 95616, USA, 2007.

[CHY96] **M. Chen, J. Han, P. Yu** - *Data Mining: An Overview from Database Perspective*, IEEE Transactions on Knowledge and Data Engineering, 8(6): 866-883, 1996.

[CK97] **T. Chia, S. Kannapan** - *Strategically Mobile Agents*, in First Intl. Workshop on Mobile Agents (MA '97), Berlin, Germany, Springer-Verlag LNCS 1219, pp. 149-161, 1997.

[G+00] **R. Gray, D. Kotz, G. Cybenko, D. Rus** - *Mobile agents: Motivations and state-of-the-art systems*, ftp://ftp.cs.dartmouth.edu/TR/TR2000-365.ps.Z., 2000.

[HPY00] **J. Han, J. Pei, Y. Yin** - *Mining frequent patterns without candidate generation*, in Proceedings of the ACM-SigMOD International Conference on Management of Data, Dallas, Texas, U.S.A., 2000.

[HGN00] **J. Hipp, U. Guntzer, G. Nakhaeizadeh** - *Algorithms for Association Rule Mining - A General Survey and Comparison*, SIGKDD Explorations, Volume 2, Issue 1, 2000.

[KHS97] **H. Kargupta, I. Hamzaoglu, B. Stafford** - *Scalable, Distributed Data Mining Using An Agent Based Architecture*, in Proc.of the 3rd Int. Conf. on Knowledge Discovery and Data Mining, Newport Beach, California, 1997.

[KIB00] **H. Kargupta, H. Ilker, S. Brian** - *Scalable, distributed data mining-an agent architecture*. In Heckerman et al. [8], page 211., 2000.

[KLZ00] **S. Krishnaswamy, S. W. Loke, A. Zaslavsky** - *Cost Models for Distributed Data Mining*. School of Computer Science &

Software Engineering, Monash University 900 Dandenong Road, Caulfield East 3145, Australia, 2000.

[O+11] **A. O. Ogunde, O. Folorunso, A. S. Sodiya, G. O. Ogunleye -** *A Review of Some Issues and Challenges in Current Agent-Based Distributed Association Rule Mining*. Asian Journal of Information Technology, Volume 10, Issue 2, Page 84-95, 2011.

[PS08] **S. Paul, P. Saravanan -** *Knowledge Integration in a Parallel and Distributed environment with association rule mining using XML data*. International Journal of Computer Science and Network Security, Vol. 8, No 5, Pg 334-339, 2008.

[Pau10] **S. Paul -** *An optimized distributed association rule mining algorithm in parallel and distributed data mining with XML data for improved response time*. International Journal of Computer Science and Information Technology, Volume 2, Number 2, April 2010. 10.5121/ijcsit.2010.2208 88

[RV10] **V. S. Rao, S. Vidyavathi -** *Distributed Data Mining and Mining Multi-Agent Data*, (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1237-1244

[Rud04] **I. Rudowsky -** *Intelligent Agents*, volume 14, pages (275-290). In Proceedings of the Communications of the Association for Information Systems, Springer, London, England, 2004.

[S+97] **S. J Stolfo, A. L Prodromidis, L. Tselepis, W. Lee, D. Fan, P. K. Chan -** *JAM: Java Agents for Meta-Learning over Distributed Databases*, in Proc. of the 3rd Int. Conf. On Data Mining and Knowledge Discovery (KDD-97), Newport Beach, California, (eds) D.Heckerman, H.Mannila, D.Pregibon, and R.Uthurusamy, AAAI Press, pp. 74-81, 1997.

[Sil06] **C. Silvestri -** *Distributed and Stream Data Mining Algorithms for Frequent Pattern Discovery*. Ph.D. Thesis: TD-2006-4, Universit`a Ca' Foscari di Venezia.

[Woo09] **M. Wooldridge -** *An Introduction to Multi-Agent Systems*. John Wiley and Sons (Chichester, United Kingdom), 2009.