

# AGENT BASED FRAMEWORKS FOR DISTRIBUTED ASSOCIATION RULE MINING: AN ANALYSIS

G. S. Bhamra<sup>1</sup>, A. K. Verma<sup>2</sup> and R. B. Patel<sup>3</sup>

<sup>1</sup>M. M. University, Mullana, Haryana, 133207 - India

<sup>2</sup>Thapar University, Patiala, Punjab, 147004- India

<sup>3</sup>Chandigarh College of Engineering & Technology, Chandigarh- 160019- India

## ABSTRACT

*Distributed Association Rule Mining (DARM) is the task for generating the globally strong association rules from the global frequent itemsets in a distributed environment. The intelligent agent based model, to address scalable mining over large scale distributed data, is a popular approach to constructing Distributed Data Mining (DDM) 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. This study performs the comparative analysis of the existing agent based frameworks for mining the association rules from the distributed data sources.*

## KEYWORDS

*Knowledge Discovery, Association Rules, Intelligent Agents, Multi-Agent System .*

## 1. INTRODUCTION

Data Mining (DM) is a process to automatically extract some interesting and valid data patterns or trends representing knowledge, implicitly stored in large databases [1], [2]. The traditional approach for knowledge discovery in distributed environment creates a single centrally integrated data repository called Data Warehouse (DW) and then DM techniques are used to mine the data and extract the knowledge [3]. The central DW based approach, however, is ineffective or infeasible because of heavy storage and computational costs involved in managing data from the ever increasing and updated distributed resources where data is produced continuously in streams. Network communication cost is also involved while transferring huge data over the wired or wireless network in a limited network bandwidth scenario. It is also not desirable to centrally collect the privacy-sensitive raw distributed data of the business organizations like banking, and telecommunication as they want only knowledge to be exchanged globally. Data from modern business organizations are not only geographically distributed but also horizontally or vertically fragmented making it difficult if not possible to combine them in a central location. Performance and scalability of a DM application can be increased by distributing the workload among sites [4].

Intelligent software agent technology is an interdisciplinary technology inherited from Distributed Computing (DC), Distributed Artificial Intelligence (DAI), advanced knowledge base systems, and human computer interaction. The motivating idea of this technology is the development and efficient utilization of autonomous software objects called agents, which have access to geographically distributed and heterogeneous information resources to simplify the complexities of DC. They are autonomous, adaptive, reactive, pro-active, social, cooperative, collaborative and flexible. They also support temporal continuity and mobility (weak and strong) within the network. An intelligent agent with mobility feature is known as Mobile Agent (MA). MA

migrates from node to node in a heterogeneous network without losing its operability. It can continue to function even if the user is disconnected from the network. It carries its code, execution state and other data while on the move. On reaching at a node MA is delivered to an Agent Execution Environment (AEE) where its executable parts are started running. Upon completion of the desired task, it delivers the results to the home node. With MA, a single serialized object is transmitted over the network carrying the small amount of resultant data only thus reducing the consumption of network bandwidth, latency (response time delay) and network traffic. They are robust, fault-tolerant and useful for low cost, light weight, portable computing devices having the low processing powers, memory constraints, and intermittent low bandwidth connection. Agent's strong mobility feature is helpful in load balancing of processor and memory intensive tasks. Number of participating hosts can be increased without any significant impact on the complexity of the application. MAs are self-contained and highly reusable. The parent agent can also clone several child agents to implement concurrent operations, and improve the efficiency. They also facilitate the rapid prototyping of distributed applications as software components can be flexibly and dynamically deployed in the form of MAs.

A Mobile Agent Platform (MAP)/Agent Execution Environment (AEE)/Agent Development Toolkit, is a middleware, distributed, server application that provides the appropriate functionality to MAs to authenticate, execute, communicate (with other agents, users, and other platforms), migrate to other platform, and use system resources in a secure way. A Multi Agent System (MAS) is distributed application comprised of multiple interacting intelligent agent components [5].

## **2. DISTRIBUTED DATA MINING**

The issues discussed above for centralized DW based DM results into the development of techniques for parallel knowledge discovery (PKD) and distributed knowledge discovery (DKD). Distributed Data Mining (DDM) is the related pattern extraction problem in DKD. DDM is concerned with application of classical DM procedures in a DC environment trying to make the best of the available resources including communication networks, computing units and distributed data repositories, human factors etc. In DDM, DM takes place both locally at each geographically distributed site and at a global level where the local knowledge is merged in order to discover global knowledge. DDM techniques are scalable where its performance does not degrade much with the increase of data set size and the number of distributed sites involved. A DDM system is a very complex entity that is comprised of many components; mining algorithms, communication subsystem, resources management, task scheduling, user interfaces, etc. It should provide efficient access to both distributed data and computing resources; monitor the entire mining procedure; and present results to users in appropriate formats. A successful DDM system is also flexible enough to adapt to various situations. It should dynamically identify the optimal mining strategy under the given resources and provide an easy way to update its components [3], [4], [6], [7], [8].

In one of the important general DDM architecture (Figure 1) proposed in [3], processing at the different distributed nodes generates several local models which are then aggregated to form a global model representing the global knowledge. Authors in [4] proposed another phase wise DDM approach. In the first phase local distributed databases are analyzed. Then, the discovered knowledge is transmitted to a merger site, where all the distributed local models are integrated. The global knowledge is then transmitted back to update the distributed databases. In some cases, instead of having a merger site, the local models are broadcasted to all other sites, so that each site can compute the global model in parallel. Data replication, data fragmentation, adaptation, interestingness property and privacy preservation of the local data are some of the issues that need to be addressed designing DDM applications [7].

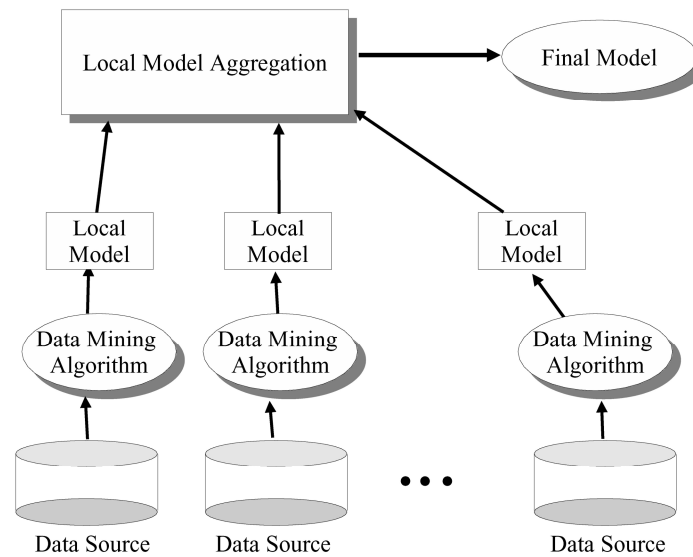


Figure 1. Distributed Data Mining Framework [3]

## 2.1. Why Agents for DDM?

Above mentioned problems and challenges of DDM and inherent features of software agents of being autonomous, capable of adaptive and deliberative reasoning clearly indicate the use of MA technology for the development of advanced DDM systems. Agent Mining or Agent enriched Data Mining or Multi Agent driven Data Mining, is an emerging interdisciplinary area that integrates MAS, DM and knowledge discovery, machine learning and other relevant areas such as statistics and semantic web. The interaction and integration between agent technology, DM and machine learning come from the intrinsic challenges, needs and opportunities faced by the constituent technologies [9]. All of the agent based DDM systems employ one or more agents per data site. These agents are responsible for analyzing local data and communicate with other agents during the mining stage. A globally coherent knowledge is synthesized via exchanges of locally mined knowledge. However, in an agent-based model, an efficient control over remote resources is inherently difficult. The motivation for the use of agent technology in DDM stems from two reasons. Firstly, there is the underlying basis of DDM being a technology that has characteristics that are intuitively suited for an agent-based approach. These characteristics are modular and well defined sub-tasks, the need to encapsulate different mining algorithms to present a common interface, the requirement for interaction and co-operation between different system and the ability to deal with the distribution. Thus, from this perspective, the focus is on the collaborating and information aspects of agency. Secondly, agent technology is seen as addressing the specific concerns of increasing scalability and enhancing performance by reducing the communication overhead associated with the transfer of large volumes of data. In the second criterion for using agents as the building blocks of DDM systems, the focus is on the mobility aspects of agency in addition to the collaborating and information aspects. Usually, such systems have one agent that acts as a controlling and coordinating entity for a task [10], [11].

## 2.2. Existing DDM Systems

There are predominantly three architectural frameworks for the development of DDM systems, the client-server model, the agent-based model and the hybrid approach which integrates the two

former techniques. The important technologies used to develop client server DDM are Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM), Enterprise Java Beans (EJB), Remote Method Invocation (RMI) and Java Database Connectivity (JDBC) [10]. The most prominent DDM systems developed using client-server architectural are Kensington Enterprise Data Mining Decision Centre [12], IntelliMiner [13] and InterAct [14].

A number of DDM solutions are provided in recent years using various techniques such as, distributed clustering, Bayesian learning, classification (regression), and compression, distributed association rules but only a few of them make use of intelligent agents [15]. The agent based model can be further classified into systems that use mobile agents and those that use stationary agents [11]. These systems are generally Java based to support the need for heterogeneity and platform independence. The most prominent DDM systems developed using agent-based architectural are PARALLEL Data Mining Agents (PADMA)[16], Java Agents for Meta-Learning (JAM) [17], Besiezing knOwledge through Distributed Heterogeneous Induction (BODHI) [18], Papyrus [19], InfoSleuth [20], Distributed Knowledge Networks (DKN) [21], a mediator oriented agent based DDM system [22], Optimized Incremental Knowledge Integration(OIKI) [23], Extendible Multi-Agent Data mining System(EMADS)[24], [25], [26], [27].

Authors in [11] compared DecisionCentre, IntelliMiner, InterAct, PADMA, JAM, BODHI, Papyrus and InfoSleuth DDM systems and proposed a hybrid model called Distributed Agent based Mining Environment (DAME) integrating the client-server and mobile agent model for delivering internet-based DDM services by incorporating cost metrics such as application run time estimation and optimization of the DDM process. Authors in [28] proposed a FIPA-compliant multi-agent platform based on mining-driven agent (Agent Academy) that offers facilities for design, implementation and deployment of multi agent systems. The researchers describe the agent academy as an attempt to develop a framework through which users can create an agent community having the ability to train and retain its own agents using DM techniques.

### 3. ASSOCIATION RULE MINING

Let  $DB = \{T_j, j = 1 \dots D\}$  be a transactional dataset of size  $D$  where each transaction  $T$  is assigned an identifier ( $TID$ ) and  $I = \{d_i, i = 1 \dots m\}$ , total  $m$  data items in  $DB$ . A set of items in a particular transaction  $T$  is called itemset or pattern. An itemset,  $P = \{d_i, i = 1 \dots k\}$ , which is a set of  $k$  data items in a particular transaction  $T$  and  $P \subseteq I$ , is called k-itemset. Support of an itemset,  $s(P) = \frac{\text{No\_of\_T\_containing\_P}}{D} \%$  is the frequency of occurrence of itemset  $P$  in  $DB$ , where No\_of\_T\_containing\_P is the support count (sup\_count) of itemset  $P$ . Frequent Itemsets (FIs) are the itemset that appear in  $DB$  frequently, i.e., if  $s(P) \geq \text{min\_th\_sup}$  (given minimum threshold support), then  $P$  is a frequent k-itemset. Finding such FIs plays an essential role in mining the interesting relationships among itemsets. Frequent Itemset Mining (FIM) is the task of finding the set of all the subsets of FIs in a transactional database. It is CPU and input/output intensive task, mainly because of the large size of the datasets involved [2].

Association Rules (ARs) first introduced in [29], are used to discover the associations (or co-occurrences) among item in a database. AR is an implication of the form  $P \Rightarrow Q$  [support,confidence] where,  $P \subset I, Q \subset I$  and  $P$  and  $Q$  are disjoint itemsets, i.e.,  $P \cap Q = \emptyset$ . An AR is measured in terms of its support and confidence factor where:

- Support  $s(P \Rightarrow Q) = p(P \cup Q) = \frac{\text{No\_of\_T\_containing\_both\_P\_and\_Q}}{D} \%$ : the probability of both  $P$  and  $Q$  appearing in  $T$ , we can say that  $s$  % of the transactions support the rule  $P \Rightarrow Q$ ,  $0 \leq s \leq 1.0$  or  $0\% \leq s \leq 100\%$ .
- Confidence  $c(P \Rightarrow Q) = p(Q|P) = \frac{s(P \Rightarrow Q)}{s(P)} = \frac{\text{sup\_count}(P \Rightarrow Q)}{\text{sup\_count}(P)} \%$ : the conditional probability of  $Q$  given  $P$ , we can say that when itemset  $P$  occurs in a transaction there are  $c$  % chances that itemset  $Q$  will occur in that transaction,  $0 \leq c \leq 1.0$  or  $0\% \leq c \leq 100\%$ .

An AR  $P \Rightarrow Q$  is said to be strong if  $s(P \Rightarrow Q) \geq \text{min\_th\_sup}$  (given minimum threshold support) and  $c(P) \geq \text{min\_th\_conf}$  (given minimum threshold confidence). Association Rule Mining (ARM) today is one of the most important aspects of DM tasks. In ARM all the strong ARs are generated from the FIs. The ARM can be viewed as two step process [30], [31].

1. Find all the frequent k-itemsets ( $L_k$ )
2. Generate Strong ARs from  $L_k$ 
  - a. For each frequent itemset,  $l \in L_k$ , generate all non empty subsets of  $l$ .
  - b. For every non empty subset  $s$  of  $l$ , output the rule " $s \Rightarrow (l-s)$ ", if  $\frac{\text{sup\_count}(l)}{\text{sup\_count}(s)} \geq \text{min\_th\_conf}$

### 3.1. Distributed Association Rule Mining

Distributed Association Rule Mining (DARM) generates the globally strong association rules from the global FIs in a distributed environment. Because of an intrinsic data skew property of the distributed database, it is desirable to mine the global rules for the global business decisions and the local rules for the local business decisions.

#### 3.1.1. Preliminaries and Definitions

Few preliminaries notations and definitions required for defining DARM and to make this study self contained are as follows:

- $S = \{S_i, i = 1 \dots n\}$ ,  $n$  distributed sites.
- $S_{CENTRAL}$ , Central Site.
- $DB_i = \{T_j, j = 1 \dots D_i\}$ , Horizontally partitioned data set of size  $D_i$  at the local site  $S_i$ , where each transaction  $T_j$  is assigned an identifier (TID).
- $DB = \bigcup_{i=1}^n DB_i$ , the aggregated dataset of size  $D = \sum_{i=1}^n D_i$ ,  $DB_i \cap DB_j = \emptyset$
- $I = \{d_i, i = 1 \dots m\}$ , total  $m$  data items in each  $DB_i$ .
- $L_{k(i)}^{FI}$ , Local frequent k-itemsets at site  $S_i$ .
- $L_{k(i)}^{FISC}$ , List of support count  $\forall Itemset \in L_{k(i)}^{FI}$ .

- $L_i^{LSAR}$ , List of locally strong association rules at site  $S_i$ .
- $L^{TLSAR} = \bigcup_{i=1}^n L_i^{LSAR}$ , List of total locally strong association rules.
- $L_k^{TFI} = \bigcup_{i=1}^n L_{k(i)}^{FI}$ , List of total frequent k-itemsets.
- $L_k^{GFI} = \bigcap_{i=1}^n L_{k(i)}^{FI}$ , List of global frequent k-itemsets.
- $L_{CENTRAL}^{GSAR}$ , List of Globally strong association rule.

Local Knowledge Base (LKB), at site  $S_i$ , comprises of  $L_{k(i)}^{FI}$ ,  $L_{k(i)}^{FISC}$  and  $L_i^{LSAR}$  which can provide reference to the local supervisor for local decisions. Global Knowledge Base (GKB), at  $S_{CENTRAL}$ , comprises of  $L^{TLSAR}$ ,  $L_k^{TFI}$ ,  $L_k^{GFI}$  and  $L_{CENTRAL}^{GSAR}$  for the global decision making. If the raw data from each of the individual databases were sent to a single database to generate the rules, certain useful rules, which would aid in making decisions about local branches, would be lost. In such cases organization may miss out certain rules that were prominent in certain branches and were not found in other branches. The frequent patterns in distributed databases are divided into three classes [32]: (a) Local patterns- Local branches need to consider the original data in their data sets so they can identify local patterns for local decisions; (b) High-vote patterns- Patterns that are supported by most of the branches and are used for making global decisions; (c) Exceptional patterns- Such patterns are strongly supported by only a few branches and used to create policies for specific branches. Like ARM, DARM task can also be viewed as two-step process [31]:

1. Find the global frequent k-itemset ( $L_k^{GFI}$ ) from the distributed Local frequent k-itemsets ( $L_{k(i)}^{FI}$ ) from the partitioned datasets.
2. Generate globally strong association rules ( $L_{CENTRAL}^{GSAR}$ ) from  $L_k^{GFI}$ .

#### 4. COMPARATIVE ANALYSIS OF EXISTING AGENT BASED DARM SYSTEMS

The existing agent based systems specifically dealing with DARM task are: Knowledge Discovery Management System (KDMS) [33], Efficient Distributed Data Mining using Intelligent Agents [34], Mobile Agent based Distributed Data Mining [35], An Agent based Framework for Association Rule Mining of Distributed Data (AFARMDD) [36], [37], Multi-Agent Distributed Association Rule Miner (MADARM) [38]. All these systems are academic research projects. Discussion of these and few others are given below.

A mobile-agent based distributed knowledge discovery system (MADKDS) is proposed in [33]. Various agents involved in the system are: Data Mining Mobile Agent (DMMA) encapsulated with a novel incremental algorithm, IAA[39] for mining the local frequent itemsets to generate the local knowledge base and return back this knowledge to Mining Process Manager, Data Pre-processing Mobile Agent (PMA) to preprocess the local data and collect back at central data warehouse which results into an increase in the storage cost at central site and Counter Mobile Agent (CMA) to scan the local databases and collect the support count of some itemsets. A central site is known as Knowledge Discovery Management System (KDMS) and distributed site are called Knowledge Discovery sub Systems (sub-KDS) in this architecture. All the mobile agents are dispatched by Mobile Agent Control Centre (MACC) at KDMS site and received and handled by Mobile Agent Execution Environment (MAEE). MACC and MAEE components on are designed on the top of IBM Aglet Workbench [44], [45]. Parallel itinerary is maintained for mobile agents and this framework is implemented using Java and C++ as dynamic link library

through Java native Interface (JNI). No privacy preserving techniques are used for the local knowledge. No user interface for the MAS is designed. No cost model for the overall DARM task is discussed and experimental validation using a large size synthetic or real data set is also required.

An Agent-Based Framework for Association Rules Mining of Distributed Data (AFARMDD) is proposed in [36], [37]. The main aim of this study is to protect the privacy of the local data from being exposed to other distributed sites encapsulating the existing techniques proposed in [42], [43] into agents. Various agents involved in the system are: Encrypt Secure Union Agent (ESUA) to encapsulate data mining operation and the encryption of Secure Union operation at each site, Decrypt secure union Agent (DSUA) to encapsulate the decryption of secure union operation at each site, Encrypt Sum Agent (ESA) to encapsulate the encryption of secure sum operation at each site, Decrypt Sum Agent (DSA) to encapsulate the decryption of secure sum operation at each site, Broadcast Agent (BA) to carry the global frequent k-itemsets to each site and Over Agent(OA) to notify all the sites at the mining operation has terminated. Parallel as well as serial itinerary is maintained for mobile agents. Agent Server and Local Host components are designed as underlying AEE. Apriori [40] algorithm is used for mining the local frequent itemsets. Privacy preserving techniques are discussed for the local sensitive data and these techniques are the core area of this study. No user interface for the MAS is designed. No cost model for the overall DARM task is discussed and experimental validation using a large size synthetic or real data set is also required. Globally strong association rules are also not generated.

Authors in [38] proposed theoretical cost models for agent based ARM in distributed data using a prototype model called Multi-Agent Distributed Association Rule Miner (MADARM). These cost models serve as a basic model to estimate and predict the response time of a DARM task. Various agents involved in the system are: Association Rule Mining Coordinating Agent (ARMCA) for creating and coordinating other agents in agent zone, Mobile Agent-Based Association Rule Miner (MAARM) for performing ARM task at each data source, Mobile Agent-Based Result Reporter (MARR) created by MAARM agent for migrating the result to ARMCA and Results Integration Coordinating Agent (RICA) for knowledge or result integration. Knowledge integration is optimized on the data sources by using agent based distributed knowledge integration (ADKI) as opposed to incremental knowledge integration proposed in [23]. Theoretical cost models are the core area of this study. Apriori [40] and FP-growth [46] algorithm are considered for mining the local frequent itemsets. Parallel itinerary for MAARM, MARR and serial itinerary for RICA agent is maintained. No underlying AEE is discussed. Only conceptual views are presented in the paper while the researchers concluded that the work still needs improvement and experimental validation.

In an experimental setup, authors in [34] performed efficient DDM intelligent agents incorporating standard Apriori [40] algorithm implemented in J#. Though authors claim that it is an agent based setup but it has been observed that there is no AEE and related agents exist in the study. So lot of work needs to be done designing an agent based framework where intelligent agents are actually implemented comprising a MAS on the top of an AEE using a large synthetic or real datasets.

Authors in [35] proposed agent based DDM approach which as an improvement over PMFI algorithm proposed in [41]. The basic objective is to reduce the time required to compute Global Frequent Itemset (GFI). The proposed algorithm performs two tasks parallel: (1) Local sites send LFIs to central site and also to all their neighbours; (2) Calculation of GFI/Candidate GFI at central site and counts of CGFI at local sites is done as an overlapped operation. That is, local sites need not wait for central site to send CGFI. Thus total time taken is reduced drastically. No information is given about which algorithm used by Mining Agent to generate Local Frequent

Itemset (LFI). No AEE exist in the study. Implementation, validation and the underlying AEE required to actually perform the agent enabled DDM using a large synthetic or real datasets. Qualitative comparison of some prominent current agent based DARM frameworks is provided in Table 1 taking into account some of the features they provide. The features include the following fields:

- *Agents* field shows the community of agents involved in the system.
- *Itinerary* indicates the serial or parallel travel plan followed by the mobile agents.
- *AEE/MAP* shows which underlying Agent Execution Environment or Mobile Agent Platform is used for developing MAS for DARM.
- *Impl* filed indicates whether the MAS is implemented along with the language used for implementation or it is just a Prototype framework.
- *Algorithm* indicates the FIM/ARM algorithm considered in the study.
- *CM* indicates whether any cost model is discussed in the study.
- *PP* points out whether any privacy-preserving mechanism is taken into account for the sensitive local data.
- *GUI* is for Graphical User Interface feature of the MAS.
- *DS* indicates whether any dataset (synthetic or real) is used in experimental validation.
- *Use* indicates the use of MAS in practical applications, development projects, case studies etc.

Table 1. Qualitative comparison of the three agent based DARM Frameworks.

Features	DARM Framework		
	MADKDS	AFARMDD	MADARM
<i>Agents</i>	DMMA, PMA, CMA	ESUA, DSUA, ESA, DSA, BA, OA	ARMCA, MAARM, MARR, RICA
<i>Itinerary</i>	Parallel	Serial and Parallel	Serial and Parallel
<i>AEE/MAP</i>	Yes	Yes	No
<i>Impl</i>	Yes	Yes	No
<i>Algorithm</i>	IAA [39]	Apriori[40]	Apriori[40], FP-growth[46]
<i>CM</i>	No	No	Yes
<i>PP</i>	No	Yes	No
<i>GUI</i>	No	No	No
<i>DS</i>	No	No	No
<i>Use</i>	No	No	No

This analysis reveals that AFARMDD [36], [37] and MADARM [38] is based on the parallel and serial itinerary of the MAs whereas MADKDS[33] use parallel itinerary. Only MADKDS[33] has an existing IBMs Aglet Workbench as the underlying AEE, others dont have any underlying AEE to test and validate the DARM system. MADARM [38] is only a prototype model without any implementation. Apriori [40] algorithm is mostly used for FIM in such systems. Only MADARM [38] discuss the cost model involved in the entire DARM task, others don't have it. Privacy preserving mechanism for the sensitive local data is the core area of AFARMDD [36], [37] system while others don't have any such mechanism. None of these frameworks has any Graphical user interface designed to work with the systems. None of these frameworks is used in any real applications, development projects, case studies etc.

Researchers in this area should focus more on developing algorithms and architecture that will reduce the massive data movement in global knowledge mining and integration thereby reducing the response time. Further algorithms and methods should also consider the development of



adaptive, fault tolerant and easily extendable systems in the area of DARM. Such systems will greatly reduce communication and interpretation costs, improve autonomy, efficiency and scalability, collaboration, security and trustworthiness of the DARM system, all of which are common issues with existing systems [8]. Agent based DARM framework must be designed on the top of an effective AEE with a GUI and implementation of all the agents involved in the system. It should effectively address and validate the cost model for the overall DARM task. Such systems must be equipped with a case study usage.

## 5. CONCLUSION

Mobile agents strongly qualify for designing distributed applications. DDM, when clubbed with the agent technology, makes a promising alliance that gives favourable results. In this study, comparative analysis of existing agent based frameworks for DARM task is done. Most of the existing agent based frameworks for DARM task are only prototype model and lacks the appropriate underlying Agent Execution Environment(AEE), scalability, privacy preserving techniques, global knowledge generation and implementation using a real datasets. With this study, we expect to contribute to cover the need of an updated review and analysis of the role of intelligent agents in designing DARM framework and also to encourage future work in this domain.

## REFERENCES

- [1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy, (1996) *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press.
- [2] J. Han & M. Kamber, (2006) *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann.
- [3] B. -H. Park & H. Kargupta, (2002) "Distributed Data Mining: Algorithms, Systems, and Applications," Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 1000 Hilltop Circle Baltimore, MD 21250, Available: [http://www.csee.umbc.edu/\\_hillol/PUBS/review.pdf](http://www.csee.umbc.edu/_hillol/PUBS/review.pdf)
- [4] G. Tsoumakas & I. Vlahavas, (2009) "Distributed Data Mining", Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, Available: <http://talos.csd.auth.gr/tsoumakas/publications/E1.pdf>
- [5] G. S. Bhamra, R. B. Patel & A. K. Verma, (2014) "Intelligent Software Agent Technology: An Overview", *International Journal of Computer Applications (IJCA)*, vol. 89, no. 2, pp. 19–31.
- [6] H. Kargupta & P. Chan, (2000) *Advances in Distributed and Parallel Knowledge Discovery*. AAAI/MIT Press.
- [7] Y. Fu, (2001) "Distributed Data Mining: An Overview", Department of Computer Science, University of Missouri-Rolla, Available: <http://academic.csuohio.edu/fuy/Pub/tcdp01.pdf>
- [8] A. O. Ogunde, O. Folorunso, A. S. Sodiya & G. O. Ogunleye, (2011) "A Review of Some Issues and Challenges in Current Agent-Based Distributed Association Rule Mining", *Asian Journal of Information Technology*, vol. 10, no. 2, pp. 84–95.
- [9] L. Cao, G. Weiss & P. S. Yu, (2012) "A brief introduction to agent mining", *Autonomous Agent and Multi-Agent System*, vol. 25, pp. 419–424.
- [10] R. Orfali, D. Harkey & J. Edwards, (1995) *The essential distributed objects survival guide*, John Wiley & Sons, USA.
- [11] S. Krishnaswami, (2002) "A Hybrid Model for Delivering Internet-based Distributed Data Mining Services", Ph.D. dissertation, School of Computer Science and Software Engineering, Monash University, Australia.
- [12] J. Chattratichat, J. Darlington, Y. Guo, S. Hedvall, M. Kohler & J. Syed, (1999) "An architecture for distributed enterprise data mining", in *High-Performance Computing and Networking*, ser. Lecture Notes in Computer Science, P. Sloot, M. Bubak, A. Hoekstra, & B. Hertzberger, Eds. Springer Berlin - Heidelberg, vol. 1593, pp. 573–582.

- [13] S. Parthasarathy & R. Subramonian, (1999) “Facilitating Data Mining on a Network of Workstations”, in *Advances in Distributed Data Mining*, H. Kargupta and P. Chan, Eds. AAAI Press, pp. 229–254.
- [14] S. Parthasarathy & S. Dwarkadas, (2002) “Shared State for Distributed Interactive Data Mining Applications”, *Journal of Distributed and Parallel Databases*, vol. 11, no. 2, pp. 129–155.
- [15] M. Klusch, S. Lodi & M. Gianluca, (2003) “The role of agents in distributed data mining: issues and benefits”, in *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology(IAT 2003)*. IEEE, pp. 211–217.
- [16] H. Kargupta, I. Hamzaoglu & B. Stafford, (1997) “Scalable, Distributed Data Mining Using An Agent Based Architecture”, in *Proceedings of the 3rd International Conference on the Knowledge Discovery and Data Mining(KDD-97)*, D. Heckerman, H. Mannila, D. Pregibon & R. Uthurusamy, Eds. AAAI Press, Menlo Park, California, pp. 211–214.
- [17] S. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan & P. K. Chan, (1997) “JAM: Java Agents for Meta-Learning over Distributed Databases”, in *Proceedings of the 3rd International Conference on the Knowledge Discovery and Data Mining(KDD-97)*, D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, Eds. AAAI Press, Menlo Park, California, pp. 74–81.
- [18] H. Kargupta, B. Park, D. Hershberger & E. Johnson, (1999) “Collective Data Mining: A new perspective toward Distributed Data Mining”, in *Advances in Distributed and Parallel Knowledge Discovery*, H. Kargupta and P. Chan, Eds. AAAI/MIT Press, pp. 131–178.
- [19] S. Bailey, R. Grossman, H. Sivakumar & A. Turinsky, (1999) “Papyrus: a system for data mining over local and wide area clusters and super-clusters”, in *Proceedings of the ACM/IEEE conference on Supercomputing (SC’99)*. ACM New York, NY, USA, p. 63.
- [20] G. L. Martin, A. Unruh & S. D. Urban, (1999) “InfoSleuth: An agent infrastructure for knowledge discovery and event detection”, *Microelectronics and Computer Technology Corporation (MCC)*, Tech. Rep. MCC-INSL-003-99.
- [21] V. Honavar, L. Miller & J. Wong, (1998) “Distributed Knowledge Networks”, in *Proceedings of the IEEE Information Technology Conference*.
- [22] S. W. Baik, J. Bala & J. S. Cho, (2005) “Agent Based Distributed Data Mining”, in *Parallel and Distributed Computing: Applications and Technologies*, ser. *Lecture Notes in Computer Science*, K.-M. Liew, H. Shen, S. See, W. Cai, P. Fan & S. Horiguchi, Eds. Springer Berlin – Heidelberg, vol. 3320, pp. 42–45.
- [23] E. I. Ariwa, M. B. Senousy & M. M. Medhat, (2003) “Informatization and E-Business Model Application for Distributed Data Mining Using Mobile Agents”, in *Proceedings of the IADIS International Conference WWW/Internet (ICWI 2003)*, pp. 85–92.
- [24] K. A. Albashiri, F. Coenen, R. Sanderson & P. Leng, (2007) “Frequent Set Meta Mining: Towards Multi-Agent Data Mining”, in *Proceedings of the 27th SGAI International Conference on Artificial Intelligence (AI 2007)*, pp. 139–151.
- [25] K. A. Albashiri & F. Coenen, (2009) “Agent-Enriched Data Mining Using an Extendable Framework”, in *Agents and Data Mining Interaction*, ser. *Lecture Notes in Computer Science*, L. Cao, V. Gorodetsky, J. Liu, G. Weiss & P. S. Yu, Eds. Springer Berlin - Heidelberg, , vol. 5680, pp. 53–68.
- [26] K. A. Albashiri, F. Coenen & P. Leng, (2009) “EMADS: An extendible multi-agent data miner”, *Knowledge-Based Systems*, vol. 22, no. 7, pp. 523–528
- [27] K. A. Albashiri, (2010) “An investigation into the issues of Multi-Agent Data Mining”, Ph.D. dissertation, The University of Liverpool, Ashton Building, Ashton Street, Liverpool L69 3BX, United Kingdom.
- [28] A. L. Symeonidis & P. A. Mitkas, (2005) *Agent Intelligence Through Data Mining*, First ed., ser. *Multiagent Systems, Artificial Societies, and Simulated Organizations*. Springer, vol. 14.
- [29] R. Agrawal, T. Imielinski & A. Swami, (1993) “Mining association rules between sets of items in large databases”, in *Proceedings of the ACM-SIGMOD International Conference of Management of Data*, pp. 207–216.
- [30] R. Agrawal & J. C. Shafer, (1996) “Parallel mining of association rules”, *IEEE Transaction on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 962–969.
- [31] M. J. Zaki, (1999) “Parallel and distributed association mining: a survey”, *IEEE Concurrency*, vol. 7, no. 4, pp. 14–25.
- [32] X. Wu & S. Zhang, (2003) “Synthesizing high-frequency rules from different data sources”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 2, pp. 353–367.

- [33] Y.-L. Wang, Z.-Z. Li & H.-P. Zhu, (2003) “Mobile agent based distributed and incremental techniques for association rules”, in Proceedings of the International Conference on Machine Learning and Cybernetics(ICMLC 2003), vol. 1, pp. 266–271.
- [34] C. Aflori & F. Leon, (2004) “Efficient Distributed Data Mining using Intelligent Agents”, in Proceedings of the 8th International Symposium on Automatic Control and Computer Science, pp. 1–6.
- [35] U. P. Kulkarni, P. D. Desai, T. Ahmed, J. V. Vadavi & A. R. Yardi, (2007) “Mobile Agent Based Distributed Data Mining”, in Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), IEEE Computer Society, pp. 18–24.
- [36] G. Hu & S. Ding, (2009a) “An Agent-Based Framework for Association Rules Mining of Distributed Data”, in Software Engineering Research, Management and Applications 2009, ser. Studies in Computational Intelligence, R. Lee and N. Ishii, Eds. Springer Berlin - Heidelberg, vol. 253, pp. 13–26.
- [37] G. Hu & S. Ding, (2009b) “Mining of Association Rules from Distributed Data using Mobile Agents,” in Proceedings of the International Conference on e-Business(ICE-B 2009), pp. 21–26.
- [38] A. O. Ogunde, O. Folorunso, A. S. Sodiya, J. A. Oguntuase & G. O. Ogunleye, (2011) “Improved cost models for agent based association rule mining in distributed databases”, *Anale SÉria Informatica*, vol. 9, no. 1, pp. 231–250, Available: <http://anale-informatica.tibiscus.ro/download/lucrari/9-1-20-Ogunde.pdf>
- [39] Y. Wang, Z. Li, J. Xue & Y. Zhao, (2002) “A Novel Incremental Algorithm for Mining Frequent Itemsets”, in Proceedings of the 2000 International Symposium on Distributed Computing and Application to Business Engineering and Science (DCABES 2002), pp. 60–64.
- [40] R. Agrawal & R. Srikant, (1994) “Fast Algorithms for Mining Association Rules in Large Databases”, in Proceedings of the 20th International Conference on Very Large Data Bases (VLDB’94). Morgan Kaufmann Publishers Inc., pp. 487–499.
- [41] Y.-L. Ruan, G. Liu & Q.-H. Li, (2005) “Parallel algorithm for Mining Frequent Item sets”, in Proceedings of the 4th International Conference on Machine Learning and Cybernetics, vol. 4. IEEE, pp. 2118–2121.
- [42] D. W. Cheung, V. T. Ng, A. W. Fu & Y. Fu, (1996) “Efficient Mining of Association Rules in Distributed Databases”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 911–922.
- [43] M. Kantarcioglu & C. Clifton, (2004) “Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1026–1037.

## AUTHORS

**Gurpreet Singh Bhamra** is currently working as Assistant Professor at Department of Computer Science and Engineering, M. M. University, Mullana, Haryana. He received his B.Sc. (Computer Sc.) and MCA from Kurukshetra University, Kurukshetra in 1995 and 1998, respectively. He is pursuing Ph.D. from Department of Computer Science and Engineering, Thapar University, Patiala, Punjab. He is in teaching since 1998. He has published 10 research papers in International/National Journals and International Conferences. He has received Best Paper Award for “An Agent enriched Distributed Data Mining on Heterogeneous Networks”, in “Challenges & Opportunities in Information Technology” (COIT-2008). He is a Life Member of Computer Society of India. His research interests are in Distributed Computing, Distributed Data Mining, Mobile Agents and Bio-informatics.



**Dr. Anil Kumar Verma** is currently working as Associate Professor at Department of Computer Science & Engineering, Thapar University, Patiala. He received his B.S., M.S. and Ph.D. in 1991, 2001 and 2008 respectively, majoring in Computer science and engineering. He has worked as Lecturer at M.M.M. Engineering College, Gorakhpur from 1991 to 1996. He joined Thapar Institute of Engineering & Technology in 1996 as a Systems Analyst in the Computer Centre and is presently associated with the same Institute. He has been a visiting faculty to many institutions. He has published over 100 papers in referred journals and conferences (India and



Abroad). He is a MISCI (Turkey), LMCSI (Mumbai), GMAIMA (New Delhi). He is a certified software quality auditor by MoCIT, Govt. of India. His research interests include wireless networks, routing algorithms and securing ad hoc networks and data mining.

**Dr. Ram Bahadur Patel** is currently working as Professor and Head at Department of Computer Science & Engineering, Chandigarh College of Engineering & Technology, Chandigarh. He received PhD from IIT Roorkee in Computer Science & Engineering, PDF from Highest Institute of Education, Science & Technology (HIEST), Athens, Greece, MS (Software Systems) from BITS Pilani and B. E. in Computer Engineering from M. M. M. Engineering College, Gorakhpur, UP. Dr. Patel is in teaching and research since 1991. He has supervised 36 M. Tech, 7 M. Phil. and 8 PhD Thesis. He is currently supervising 6 PhD students. He has published 130 research papers in International/National Journals and Refereed International Conferences. He has written 7 text books for engineering courses. He is member of ISTE (New Delhi), IEEE (USA). He is a member of various International Technical Committees and participating frequently in International Technical Committees in India and abroad. His current research interests are in Mobile & Distributed Computing, Mobile Agent Security and Fault Tolerance and Sensor Network.

