International Journal of Reviews in Computing

IJRIC

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

MULTI AGENT-BASED DISTRIBUTED DATA MINING: AN OVER VIEW

VUDA SREENIVASA RAO

Research Scholar,CSIT Department,JNT University , Hyderabad. Andhra Pradesh, INDIA

E-mail: vudasrinivasarao@gmail.com

ABSTRACT

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. The Data Mining technology normally adopts data integration method to generate Data warehouse,on which to gather all data into a central site, and then run an algorithm against that data to extract the useful Module Prediction and knowledge evaluation. However, a single data-mining technique has not been proven appropriate for every domain and data set. Data mining techniques involving in such complex environment must encounter great dynamics due to changes in the system can affect the overall performance of the system. Agent computing whose aim is to deal with complex systems has revealed opportunities to improve distributed data mining systems in a number of ways. Multi-agent systems (MAS) often deal with complex applications that require distributed problem solving. In many applications the individual and collective behavior of the agents depends on the observed data from distributed sources. Distributed data mining is originated from the need of mining over decentralized data sources. The field of Distributed Data Mining (DDM) deals with these challenges in analyzing distributed data and offers many algorithmic solutions to perform different data analysis and mining operations in a fundamentally distributed manner that pays careful attention to the resource constraints. Since multi-agent systems are often distributed and agents have proactive and reactive features which are very useful for Knowledge Management Systems, combining DDM with MAS for data intensive applications is appealing.

This paper the integration of multi-agent system and distributed data mining, also known as multi agent-based distributed data mining, in terms of significance, system overview, existing systems, and research trends.

Key words: *Distributed Data Mining, Multi-Agent Systems, Multi Agent Data Mining, Agent Based Distributed Data Mining.*

1. INTRODUCTION:

Originated from knowledge discovery from databases (KDD), also known as data Mining (DM), distributed data mining (DDM) mines data sources regardless of their physical locations. The need for such characteristic arises from the fact that data produced locally at each site may not often be transferred across the network due to the Excessive amount of data and privacy issues. Recently, DDM has become a critical components of knowledge-based systems because its decentralized architecture reaches every networked business.

Data Mining still poses many challenges to the research community. The main challenges in data mining are:

- Data mining to deal with huge amounts of data located at different sites The amount of data can easily exceed the terabyte limit;
- Data mining is very computationally intensive process involving very large data sets. Usually, it is necessary to partition and distribute the data for parallel processing to achieve acceptable time and space performance;
- Input data change rapidly. In many application domain data to be mined either is produced with high rate or they actually come in streams. In those cases, knowledge has to be mined fast and efficiently in order to be usable and updated;

Security is a major concern in that companies or other organizations may be willing to release data mining results but not the source data itself.

The remaining sections of the paper are organized as follows. In Section II we describe

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

the agent. In Section III we describe Agent-Based Distributed Data Mining. In Section IV we describe open problems Strategy Section V MADM systems general Architecture over view .Section VI Open Issues and Trends and cludes the paper.

2. WHY AGENTS:

DDM is a complex system focusing on the distribution of resources over the network as well as data mining processes. The very core of DDM systems is the scalability as the system configuration may be altered time to time, therefore designing DDM systems deals with great details of software engineer issues, such re-usability, extensibility, and robustness. For these reasons, agents' characteristics are desirable for DDM systems.Furthermore, the decentralization property seems to fit best with the DDM requirement. At each data site, mining strategy is deployed specifically for the certain domain of data. However, there can be other existing or new strategies that data miner would like to test. A data site should seamlessly integrate with external methods and perform testing on multiple strategies for further analysis. Autonomous agent can be treated as a computing unit that performs multiple tasks based on a dynamic configuration. The agent interprets the configuration and generates an execution plan to complete multiple tasks.[2], [21], [14], [20], [27], and [26] discuss the benefits of deploying agents in DDM systems. Nature of MAS is decentralization and therefore each agent has only limited view to the system. The limitation somehow allows better security as agents do not need to observe other irrelevant surroundings. Agents, in this way, can be programmed as compact as possible, in which light-weight agents can be transmitted across the network rather than the data which can be more bulky. Being able to transmit agents from one to another host allows dynamic organization of the system. For example, mining agent $a1$, located at site $s1$, posses algorithm $alg1$. Data mining task $t1$ at site $s2$ is instructed to mine the data using $alg1$. In this setting, transmitting $a1$ to $s2$ is a probable way rather than transfer all data from $s2$ to $s1$ where $alg1$ is available.

In addition, security, a.k.a. trust-based agents[41][19], is a critical issue in ADDM. Rigid security models intending to ensure the security may degrade the system scalability. Agents offer alternative solutions as they can travel through the system network. As in [29], the authors present a framework in which mobile agents travel in the system network allowing the system to maintain data privacy. Thanks to the self-organization characteristic which excuses the system from transferring data across the network therefore adds up security of the data .

A trade-off for the previous discussed issue, scalability is also a critical issue of a distributed system. In order to inform every unit in the system about the configuration update of the system, such as a new data site has joined the system, demands extra human interventions or high complex mechanism in which drops in performance may occur. To this concern, collaborative learning agents[40][6] are capable of sharing information, in this case, about changes of system configuration, and propagate from one agent to another allowing adaptation of the system to occur at individual agent level. Furthermore, mobile agents as discussed earlier can help reduce network and DM application server load as in state-of-art systems[3][24].

3. AGENT-BASED DISTRIBUTED DATA MINING:

Applications of distributed data mining include credit card fraud detection system,intrusion detection system, health insurance, security-related applications, distributed clustering, market segmentation, sensor networks, customer profiling, evaluation of retail promotions, credit risk analysis, etc. These DDM application can be further enhanced with agents. ADDM takes data mining as a basis foundation and is enhanced with agents; therefore, this novel data mining technique inherits all powerful properties of agents and, as a result, yields desirable characteristics. In general, constructing an ADDM system concerns three key characteristics: interoperability, dynamic system configuration, and performance aspects, discussed as follows.Interoperability concerns, not only collaboration of agents in the system, but also external interaction which allow new agents to enter the system seamlessly. The architecture of the system must be open and flexible so that it can support the interaction including communication protocol, integration policy, and service directory. Communication protocol covers message encoding, encryption, and transportation between agents, nevertheless, these are standardized by the Foundation of Intelligent Physical Agents (FIPA)1 and are available for public access. Most agent platforms, such as JADE2 and JACK3, are FIPA

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

compliant therefore interoperability among them are possible. Integration policy specifies how a system behaves when an external component, such as an agent or a data site, requests to enter or leave. The issue is further discussed in [47] and [29] In relation with the interoperability characteristic, dynamic system configuration, that tends to handle a dynamic configuration of the system, is a challenge issue due to the complexity of the planning and mining algorithms. A mining task may involve several agents and data sources, in which agents are configured to equip with an algorithm and deal with given data sets. Change in data affects the mining task as an agent may be still executing the algorithm.Lastly, performance can be either improved or impaired because the distribution of data is a major constraint. In distributed environment, tasks can be executed in parallel, in exchange, concurrency issues arise. Quality of service control in performance of data mining and system perspectives is desired, however it can be derived from both data mining and agents fields.

Next, we are now looking at the overview of our point of focus. An ADDM system can be generalized into a set of components and viewed as depicted in figure 1.We may generalize activities of the system into request and response, each of which involves a different set of components. Basic components of an ADDM system are as follows.

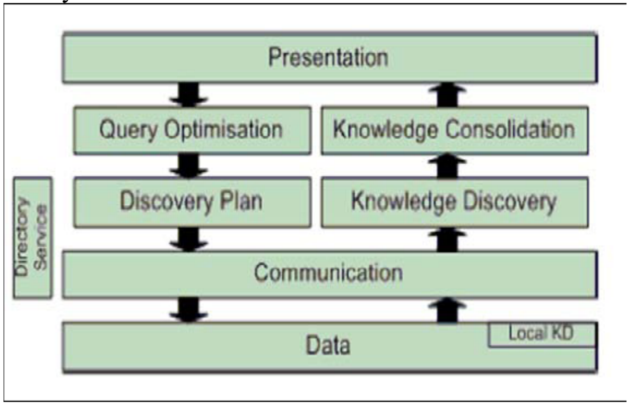

Fig. 1: Overview of ADDM system.

Data: Data is the foundation layer of our interest. In distributed environment, data can be hosted in various forms, such as online relational databases, data stream, web pages, etc., in which purpose of the data is varied.

Communication: The system chooses the related resources from the directory service, which maintains a list of data sources, mining algorithms, data schemas, data types, etc. The communication protocols may vary depending on implementation of the system, such as client-server, peer-to-peer, etc.

Presentation: The user interface (UI) interacts with the user as to receive and respond to the user. The interface simplifies complex distributed systems into user-friendly message such as network diagrams, visual reporting tools, etc. On the other hand, when a user requests for data mining through the UI, the following components are involved.

Query optimization: A query optimizer analyses the request as to determine type of mining tasks and chooses proper resources for the request. It also determines whether it is possible to parallelize the tasks, since the data is distributed and can be mined in parallel.

Discovery Plan: A planner allocates sub-tasks with related resources. At this stage, mediating agents play important roles as to coordinate multiple computing units since mining sub-tasks performed asynchronously as well as results from those tasks. On the other hand, when a mining task is done, the following components are taken place,

Local Knowledge Discovery (KD): In order to transform data into patterns which adequately represent the data and reasonable to be transferred over the network, at each data site, mining process may take place locally depending on the individual implementation.

Knowledge Discovery: Also known as mining, it execute the algorithm as required by the task to obtain knowledge from the specified data source.

Knowledge Consolidation: In order to present to the user with a compact and Meaningful mining result, it is necessary to normalize the knowledge obtained from various sources. The component involves a complex methodologies to combine knowledge/patterns from distributed sites. Consolidating homogeneous knowledge/patterns is promising and yet difficult for heterogeneous case.

4. OPEN PROBLEMS STRATEGY:

Several systems have been developed for distributed data mining. These systems can be classified according to their strategy to three types; central learning, meta-learning, and hybrid learning.

4.1 Central learning strategy: is when all the data can be gathered at a central site and a single

International Journal of Reviews in Computing

IJRIC

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

model can be build. The only requirement is to be able to move the data to a central location in order to merge them and then apply sequential DM algorithms. This strategy is used when the geographically distributed data is small. The strategy is generally very expansive but also more accurate. The process of gathering data in general is not simply a merging step; it depends on the original distribution. For example, different records are placed in different sites, different attributes of the same records are distributed across different sites, or different tables can be placed at different sites, therefore when gathering data it is necessary to adopt the proper merging strategy. However, as pointed before this strategy in general is unfeasible [10]. Agent technology is not very preferred in such strategy.

4.2 Meta-learning strategy: it offers a way to mine classifiers from homogeneously distributed data. Meta-learning follows three main steps. The first is to generate base classifiers at each site using a classifier learning algorithms. The second step is to collect the base classifiers at a central site, and produce meta-level data from a separate validation set and predictions generated by the base classifier on it. The third step is to generate the final classifier (meta-classifier) from meta-level data via a combiner or an arbiter. Copies of classifier agent will exist or deployed on nodes in the network being used. Perhaps the most mature systems of agent-based meat-learning systems are: JAM system [11], and BODHI [11].

4.3 Hybrid learning strategy: is a technique that combines local and centralized learning for model building [12]; for example, Papyrus [13] is designed to support both learning strategies. In contrast to JAM and BODHI, Papyrus can not only move models from site to site, but can also move data when that strategy is desired. Papyrus is a specialized system which is designed for clusters while JAM and BODHI are designed for data classification.

The major criticism of such systems is that it is not always possible to obtain an exact final result, i.e. the global knowledge model obtained may be different from the one obtained by applying the one model approach (if possible) to the same data.

Approximated results are not always a major concern, but it is important to be aware of that. Moreover, in these systems hardware resource usage is not optimized. If the heavy computational part is always executed locally to data, when the same data is accessed concurrently, the benefits coming from the distributed environment might vanish due to the possible strong performance degradation. Another drawback is that occasionally, these models are induced from databases that have different schemas and hence are incompatible.

5. MADM SYSTEMS GENERAL ARCHITECTURE OVER VIEW:

In distributed data mining, there is a fundamental trade-off between the accuracy and the cost of the computation. If our interest is in cost functions which reflect both computation costs and communication costs, especially the cost of wide area communications, we can process all the data locally obtaining local results, and combine the local results at the root to obtain the final result. But if our interest is accurate result, we can ship all the data to a single node. We assume that this produces the most accurate result. In general, this is the most expensive while the former approach is less expensive, but also less accurate.

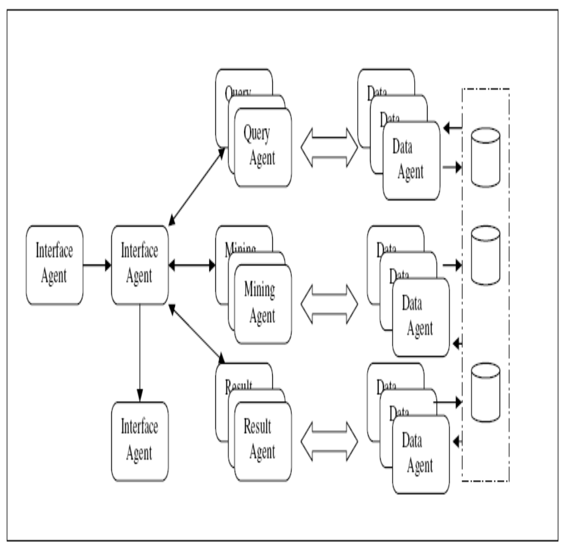

Fig 2: MADM systems general Architecture.

Architecture:

Most of the MADM frameworks adapt similar architecture (see figure .2.) and provide common structural components [48],[49],[50]. They use KQML or FIPA-ALC, which are a standard agent communication language that facilitates the interactions among agents. The following is a

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

definition for the most common agents that are used in MADM; the names might be different but they share the same functionalities in most cases.

Interface Agent (or User Agent): this agent interacts with the user (or user agent). It asks the user to provide his requirements, and provides the user with mined results (may be visualized). Its interface module contains methods for inter agent communication and getting input from the user. The process module contains methods for capturing the user input and communicating it to the facilitator agent. In the knowledge module, the agent stores the history of user interaction, and user profiles with their specific preferences.

Facilitator Agent (or Manager Agent): the facilitator agent is responsible of the activation and synchronization of different agents. It elaborates a work plan and is in charge of ensuring that such a work plan is fulfilled. It receives the assignments from the interface agent and may seek the services of a group of agents and synthesize the final result and present it to the interface agent. The interface module is responsible for interagent communication; the process module contains methods for control and coordination of various tasks. The sequence of tasks to be executed is created from specific "ontologies" stored in the knowledge module using a rule-based approach. The agent task may include identifying relevant data sources, requesting services from agents,
generating queries, etc. The knowledge module also contains meta-knowledge about capabilities of other agents in the system.

Resource Agent (or Data Agent): The resource agent actively maintains meta-data information about each of the data sources. It also provides predefined and ad hoc retrieval capabilities. It is responsible for retrieving the necessary data sets requested by the data mining agent in preparation for a specific data mining operation. It takes into account the heterogeneity of the databases, as well as resolves conflicts in data definition and representation. Its interface module supports inter-agent communication as well as interface to existing data sources. The process module provides facilities for ad hoc and predefined data retrieval. Based on the user request, appropriate queries are generated and executed against the data base and the results are communicated back to the facilitator agent, or other agents.

Mining Agent: The data mining agent implements specific data mining techniques and algorithms. The interface module supports inter-agent communication. The process module contains methods for initiating and carrying out the data mining activity, capturing the results of data mining, and communicating it to result agent or the facilitator agent. The knowledge module contains meta-knowledge about data mining methods, i.e., what method is suitable for what type of problem, input requirements for each of the mining methods, format of input data, etc. This knowledge is used by the process module in initiating and executing a particular data mining algorithm for the problem at hand.

Result Agent: Result agent observes a movement of mining agents, and obtains result from mining agents. When result agent obtains all results, it arrangement/integrates with the facilitator agent to show the result to the user. The interface module may provide access to other visualization software that may be available within the organization. The process module contains methods to support *ad hoc* and predefined reporting capabilities, generating visual representations, and facilitating user interaction. The knowledge module stores details about report templates and visualization primitives that can be used to present the result to the user.

Broker Agent (or Matchmaker Agent): the broker *agent* serves as an advisor agent that facilitates the diffusion of requests to agents that have expressed an ability to handle them. This is performed by accepting advertisements from supply facilitators and recommendation requests from request facilitators. It keeps track of the names, ontology, and capabilities of all registered agents in the system; it can reply to the query of an agent with the name and ontology of an appropriate agent that has the capabilities requested. In general, any new agents in a system using a Broker Agent must advertise their capabilities through the broker in order to become a part of the agent system (yellow pages service).

Query Agent: Query agent is generated at each demand of a user. The knowledge module contains meta-data information including the local schemas and a global schema. These schemas are used in generating the necessary queries for data retrieval.

Ontology Agent: maintains and provides overall knowledge of ontologies and answers queries about the ontologies. It may simply store the ontology as given, or it may be as advanced as to be able to use semantic reasoning to determining

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

the applicability of a domain to any particular data mining request.

Mobile Agent: some systems use the agent mobility feature. A mobile agent travels around the network. On each site, it processes the data and sends the results back to the main host, instead of expensive transferring large amount of data across the network. This has the advantage of low network traffic because the agents do data processing locally. However, it provokes a major security issues. As an organization receiving a mobile agent for execution at your local machine require strong assurances about the agent's attentions. There is also the requirement of installing agent platform at each site.

Local Task Agent: in most of the system the Data Agent is a local agent located at the local site. It can submit its information to the facilitator agent, it can also response to data mining requests of mining agents. A local agent can retrieve its local database, performs calculations and returns its results to the system.

KDD system agents:

Some MADM systems contain other agents to maintain the whole process of the knowledge discovery in data which include data preparation and data evolution. These agents are:

Pre-processing Agent: It prepares data for mining. It is responsible for performing the necessary data cleansing before using the data set for data mining. The process module contains methods for data cleansing and data preparation needed for specific data mining algorithms

Post data mining Agent: it evaluates the performance and accuracy, etc., of data mining agents.

6. OPEN ISSUES AND TRENDS:

The interaction and integration between the two technologies have explore the new challenges. Considering various ingredients for the integration could be a key to rapidly enhance the development process and usability of the system, let us examine them from different perspectives. Research Perspective: Data distribution in real-life applications are either homogeneous or heterogeneous. Data can be partitioned both vertically and horizontally, and furthermore data splitting may not be available across the sites. For examples, two related customer databases may not reflect each others in which a customer may never provide contact details but somehow appear to buy some products. The applications will require a data mining technology to pay careful attention to the distributed computing, communication, and storage of the system. Another approach to develop MADDM is an inspiration from the nature which has proven to be promising. Swarm intelligence is closely related to intelligent agents.

Recently, researchers pay attention to the possibility to implement DDM systems with swarm intelligence. Sample applications of swarm intelligence in data mining are rule-based classifiers using ants, feature selection with ant colony optimisation, data and text mining with hierarchical clustering ants, etc. Further readings can be found in [1] and [38]. Software Engineering Perspective: Expectedly, MADDM frequently requires exchange of data mining models among the data sites. Therefore, seamless and transparent realisation of DDM technology will require standardised schemes to represent and exchange models. Therefore, software engineering tools that support the design of data mining and distributed database are desired. So far, PMML, the Cross-Industry Standard Process Model for Data mining (CRISP-DM), and other related efforts are likely to be very useful. The very basic foundation of our focus is the database. Not only full-scale database, like relational database, is taken into consideration during system integration. Desktop and light-weight database running on limited devices, such as mobile phones, can be integrated into ADDM. Mobile agents can be migrated (downloaded) and perform task on the devices and take back only a representative model for further analysis. The second ingredient is the emergence of service oriented architecture (SOA) that enables agent-based application to integrate better than ever. SOA is a promising architecture as it is widely adapted in several applications. We cannot deny the fact that web-based applications are becoming more and more popular. Internet has become a necessary element of a computer system. System Perspective: A novel very perspective but poorly researched application area of agents and data mining synergy is mobile, ubiquitous and peer-to-peer (P2P) computing. A specific feature of such computing systems is that the latter operate with dynamic set of information sources. E.g., the mobile devices may move and freely enter to and exit from the network thus changing the set of network nodes and communication topology, changing the set of available services as well. Examples of such application areas are, e.g., smart space and ambient intelligence. In these

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

environments, decisions are made on the basis of fusion of information received from distributed sensors and mobile devices populating the environment. One of the objectives of such application is adaptation to multiple human habits that can be achieved through learning of multiple human profiles. On the other hand, for class of applications in question, multi-agent approach supplies for most natural framework, appropriate architecture, as well as design technology. Thus, integrating agent and data mining in ubiquitous environments like smart space, ambient intelligence, etc., could be very perspective and promising to reach high quality performance of corresponding applied systems. In fact, ubiquitous and mobile computing form a novel and very perspective, although poorly researched, application area of agents and data mining synergy. A specific feature of such computing systems is that the latter often has to handle with dynamic set of information sources. E.g., the mobile devices may move and freely enter to and exit from the network thus changing the set of network nodes and communication topology, changing the set of available services as well. Examples of such application areas are, e.g., smart space and ambient intelligence. In these environments, decisions are made on the basis of fusion of information received from distributed sensors and mobile devices populating the environment. One of the objectives of such application is adaptation to multiple human habits that can be achieved through learning of multiple human profiles. On the other hand, for class of applications in question, multi-agent approach supplies for most natural framework, appropriate architecture, as well as sound design technology. Thus, integrating agent and data mining in ubiquitous environments like smart space, ambient intelligence, etc., could be very perspective and promising to reach high qual ty performance of corresponding applied systems. [15] presents a summary of challenges integrating ubicomp with MAS for data mining task. Recently, peer-to-peer (P2P) computing has proven its excellence through its product, such as peer download software, file sharing software, which they gather users to join the service quickly. P2P is respected as one of the best scalable system, and thus it increases availability of the system as millions of peers can be attached to the network. P2P algorithm does not rely on a central server, each unit performs its own task and requests for data from others if available in order to save the redundant time. However, security is a critical issue in P2P due to exchanging information with other peers that can add a vulnerability to the network, such as denial of service or selfish behaviour. Some peers may only consume others' resources while they do

not provide to others. Nevertheless, each peer must agree of terms and conditions of use before joining the network. P2P has caught researchers attention due to the compliance with multi-agent systems as appear in [18] and [11]. User Perspective: Finally human-computer interaction issues in DDM offers some unique challenges. It requires system-level support for group interaction, collaborative problem solving, development of alternate interfaces (particularly for mobile devices), and dealing with security issues.

7. DM TASK PLANNING AND FLOW OF SYSTEM OPERATIONS:

DM task planning is realized by negotiation between the facilitator agent and mining agents through the message passing mechanism. Suppose a user agent sends a request to the facilitator agent to inform that it would like to do data mining with other agents in the organization. The user agent also needs to give information of model definition (dependent and independent attributes, attribute type (numeric or categorical), model type (linear or nonlinear) with its request. When the facilitator receives the request from the user agent, it negotiates with the broker agent to determine which agents to launch for this task. For example, if the user wants all possible rules meeting minimum support and confidence levels, across all available data sources, then the mining agent must ask every data agent possible for all association rules. If the user wants to find all items Y which have statistical significance for a given X, the mining agent must ask only those agents which have information about X. It would be time consuming and wasteful to ask agents which have access to data not containing X, as no rules would be generated.Finally, if the user specifies a X and a Y and asks for the level of support and confidence between the two, the agent must again only ask those agents that have information about both X and Y. The mining agent is then responsible for completing the task, while the facilitator agent continues to plan future DM requests. When the mining agent is

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

completed it returns the results and the facilitator agent passes them onto the user agent.

8. SUMMARY:

In this document I presented an overview of MADM systems that are exist today.
I defined the common components between these systems and give a description to their strategies and architecture.

REFERENCES:

[1]. Ajith Abraham, Crina Grosan, and Vitorino Ramos, editors. *Swarm Intelligence in Data Mining*, volume 34 of *Studies in Computational Intelligence*. Springer, 2006.

[2]. Sung W. Baik, Jerzy W. Bala, and Ju S. Cho. Agent based distributed data mining. *Lecture Notes in Computer Science*, 3320:42–45, 2004.

[3]. S. Bailey, R. Grossman, H. Sivakumar, and A. Turinsky. Papyrus: a system for data mining over local and wide area clusters and super-clusters. In *Supercomputing '99: Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM)*, page 63, New York, NY, USA, 1999. ACM.

[4]. R. J. Bayardo, W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, and Others. InfoSleuth: agent-based semantic integration of information in open and dynamic environments. *ACM SIGMOD Record*, 26(2):195–206, 1997.

[5]. F. Bergenti, M. P. Gleizes, and F. Zambonelli. *Methodologies And Software Engineering For Agent Systems: The Agent-oriented Software Engineering Handbook*. Kluwer Academic Publishers, 2004.

[6]. A. Bordetsky. Agent-based Support for Collaborative Data Mining in Systems Management. In *Proceedings Of The Annual Hawaii International Conference On System Sciences*, page 68, 2001.

[7]. R. Bose and V. Sugumaran. IDM: an intelligent software agent based data mining environment. *1998 IEEE International Conference on Systems, Man, and Cybernetics*, 3, 1998.

[8]. L. Cao, C. Luo, and C. Zhang. Agent-Mining Interaction: An Emerging Area. *Lecture Notes in Computer Science*, 4476:60, 2007.

[9]. L. Cao, J. Ni, J. Wang, and C. Zhang. Agent Services-Driven Plug and Play in the F-TRADE. In *17th Australian Joint Conference on Artificial Intelligence*, volume 3339, pages 917–922. Springer, 2004.

[10]. J. Dasilva, C. Giannella, R. Bhargava, H. Kargupta, and M. Klusch. Distributed data mining and agents. *Engineering Applications of Artificial Intelligence*, 18(7):791–807, October 2005.

[11]. S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed data mining in peer-to-peer networks. *Internet Computing, IEEE*, 10(4):18–26, 2006.

[12]. W. Davies and P. Edwards. Distributed Learning: An Agent-Based Approach to Data-Mining. In *Proceedings of Machine Learning 95 Workshop on Agents that Learn from Other Agents*, 1995.

[13]. U. Fayyad, R. Uthurusamy, and Others. Data mining and knowledge discovery in databases. *Communications of the ACM*, 39(11):24–26, 1996.

[14]. C. Giannella, R. Bhargava, and H. Kargupta. Multi-agent Systems and Distributed Data Mining. *Lecture Notes in Computer Science*, pages 1–15, 2004.

[15]. V. Gorodetskiy. Interaction of agents and data mining in ubiquitous environment. In *Proceedings of the 2008 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08)*, 2008.

[16]. V. Gorodetsky, O. Karsaev, and V. Samoilov. Multi-Agent Data and Information Fusion. *Nato Science Series Sub Series Iii Computer And Systems Sciences*, 198:308, 2005.

[17]. V. Gorodetsky, O. Karsaev, and V. Samoilov. Infrastructural Issues for Agent-Based Distributed Learning. In *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pages 3–6. IEEE Computer SocietyWashington, DC, USA, 2006.

[18]. V. Gorodetsky, O. Karsaev, V. Samoylov, and S. Serebryakov. P2P Agent Platform: Implementation and Testing. In *Proceedings International Workshop "Agent and Peer-to-Peer Computing"(AP2PC-2007) associated with AAMAS-07. Honolulu, Hawaii*, pages 21–32, 2007.

[19]. V. Gorodetsky and I. Kotenko. The Multi-agent Systems for Computer Network Security Assurance: frameworks and case studies. In *IEEE International Conference on*

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

Artificial Intelligence Systems, 2002, pages 297–302, 2002.

[20]. Vladimir Gorodetsky, Oleg Karsaev, and Vladimir Samoilov. Multi-agent technology for distributed data mining and classification. In *IAT*, pages 438–441. IEEE Computer Society, 2003.

[21]. Sven A. Brueckner H. Van Dyke Parunak. Engineering swarming systems. *Methodologies and Software Engineering for Agent Systems*, pages 341–376, 2004.

[22]. H. Kargupta, I. Hamzaoglu, and B. Stafford. Scalable, distributed data mining using an agent based architecture. In *Proceedings the Third International Conference on the KnowledgeDiscovery and Data Mining, AAAI Press, Menlo Park, California*, pages 211–214, 1997.

[23]. H. Kargupta and K. Sivakumar. Existential pleasures of distributed data mining. *Data Mining:Next Generation Challenges and Future Directions*, pages 1–25, 2004.

[24]. Hillol Kargupta, Byung-Hoon Park, Daryl Hershberger, and Erik Johnson. Collective data mining: A new perspective toward distributed data mining. In *Advances in Distributed and Parallel Knowledge Discovery*, pages 133–184, 1999.

[25]. M. Klusch, S. Lodi, and G. Moro. Agent-based distributed data mining: The KDEC scheme. *AgentLink, Springer Lecture Notes in Computer Science*, 2586, 2003.

[26]. M. Klusch, S. Lodi, and G. Moro. The role of agents in distributed data mining: Issues and benefits. In *Proceedings of the 2003 IEEE/WIC International Conference on Intelligent AgentTechnology (IAT 2003)*, pages 211–217, 2003.

[27]. Matthias Klusch, Stefano Lodi, and Gianluca Moro. Issues of agent-based distributed data mining. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, pages 1034–1035. ACM, 2003.

[28]. Raymond S. T. Lee and James N. K. Liu. ijade web-miner: An intelligent agent framework for internet shopping. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):461–473, 2004.

[29]. Xining Li and Jingbo Ni. Deploying mobile agents in distributed data mining. *Lecture Notes in Computer Science*, 4819:322–331, 2007.

[30]. F. Menczer, R. K. Belew, and W. Willuhn. Artificial life applied to adaptive information agents. In *AAAI Spring Symposium on Information Gathering*, pages 128–132, 1995.

[31]. S. Merugu and J. Ghosh. A distributed learning framework for heterogeneous data sources. In *Conference on Knowledge Discovery in Data*, pages 208–217. ACM Press New York, NY, USA, 2005.

[32]. Peerapol Moemeng, Longbing Cao, and Chengqi Zhang. F-TRADE 3.0: An Agent-Based Integrated Framework for Data Mining Experiments. In *Proceedings of IEEE/WIC/ACM International Conference onWeb Intelligence and Intelligent Agent Technology*, volume 3, pages 612–615, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[33]. L. Panait and S. Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[34]. Byung H. Park and Hillol Kargupta. Distributed data mining: Algorithms, systems, and applications. In Nong Ye, editor, *Data Mining Handbook*, pages 341–358. Lawrence Earlbaum Associates, 2002.

[35]. M. Penang. Distributed Data Mining From Heterogeneous Healthcare Data Repositories: Towards an Intelligent Agent-Based Framework. In *Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems:(CBMS 2002): 4-7 June, 2002, Maribor, Slovenia*. IEEE Computer Society, 2002.

[36]. Andreas Leonidas Prodromidis. *Management of intelligent learning agents in distributed data mining systems*. PhD thesis, Columbia University, New York, NY, USA, 1999.

[37]. S. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, and P. K. Chan. JAM: Java agents for meta-learning over distributed databases. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 74–81, 1997.

[38]. H. Van Dyke Parunak Sven A. Brueckner. Swarming distributed pattern detection and classification. In *Environments for Multi-Agent Systems*, pages 232–245. Springer, 2005.

[39]. J. Tozicka, M. Rovatsos, and M. Pechoucek. A framework for agent-based distributed machine learning and data mining. In *Proceedings of the 6th international joint conference on Autonomous agents and*

multiagent systems. ACM New York, NY, USA, 2007.
[40]. Jan Tozicka, Michal Jakob, and Michal Pechoucek. Market-inspired approach to collaborative learning. *Lecture Notes in Computer Science*, 4149:213–227, 2006.
[41]. Frank E. Walter, Stefano Battiston, and Frank Schweitzer. A model of a trust-based recommendation system on a social network, Nov 2006.
[42]. Gerhard Weiss. A multiagent perspective of parallel and distributed machine learning. In *Agents*, pages 226–230, 1998.
[43]. M.Wooldridge and N. R. Jennings. Agent Theories, Architectures, and Languages: A Survey. *Intelligent Agents*, 22, 1995.
[44]. M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
[45]. S. Z. H. Zaidi, S. S. R. Abidi, S. Manikam, and Cheah Yu-N. Admi: a multi-agent architecture to autonomously generate data mining services. In *Intelligent Systems, 2004. Proceedings.2004 2nd International IEEE Conference*, volume 1, pages 273–279 Vol.1, 2004.
[46]. Mohammed J. Zaki. Parallel and distributed association mining: A survey. *IEEE Concurrency*, 7(4):14–25, 1999.
[47]. Gorodetskiy, V.; Karsaev, O.; Samoilov, V.; Serebryakov, S., Agent-based Service-Oriented Intelligent P2P Networks for Distributed Classification *Hybrid Information Technology*, 2006.
[48]. [48] H.Kargupta, I.Hamzaoglu and B.Stafford "Scalable, Distributed Data Mining An Agent Based Application".Proceedings of Knowledge Discovery And Data Mining, August, 1997.
[49]. [49]Kargupta, H., Hamzaoglu, I., Stafford, B.: Scalable, distributed data mining using an agent-based architecture. In D.Heckerman, H.Mannila, D.Pregibon, R.Uthurusamy, eds.: Proc. 3rd International Conference on Knowledge Discovery and Data Mining, Newport Beach, California, USA, AAAI Press (1997) 211–214
[50]. [50] Kargupta, H., Park, B., Hershberger, D., Johnson, E.: 5, Collective Data Mining: A New Perspective Toward Distributed Data Mining. In: Advances in Distributed and
[51]. Parallel Knowledge Discovery. AAAI/MIT Press (2000) 131–178

AUTHOR PROFILE:

Vuda .Srinivasarao received the M.Tech degree in Computer Science & Engg from the Satyabama University, in 2007. . He is research scholar in CSIT Department, JNT University Hyderabad Andhra Pradesh, India. His research interests include Network Security, Cryptography, Data Mining & Artificial Intelligence.