Approaches and Techniques of Distributed Data Mining : A Comprehensive Study

S.Urmela[1], Dr.M.Nandhini[2]
[1]Research Scholar, [2]Assistant Professor
[1,2] Department of Computer Science, Pondicherry University, India
[1] urmelaindra@gmail.com
[2] mnandhini2005@yahoo.com

***Abstract* - Distributed Data Mining (DDM) has become one of the promising areas of Data Mining (DM). DDM evolved from DM from the urge to mine data from distributed sites. DM paved way for increased computational cost and privacy due to centralized data mining, whereas DDM paved way for decrease in computational cost as well as enhanced data privacy by distributing resources across distributed sites. Mining techniques framed for DM can't be applied for DDM since mining DDM follows a different strategy compared to DM. DDM includes classifier based, agent based and privacy preserving based approaches. In this paper, DDM approaches and techniques is studied in detail.**

Keyword - Distributed Data Mining, distributed sites, computation cost, classifier approach, agent based, privacy-preserving

I. INTRODUCTION

Data Mining (DM) is the process of extracting useful information from datasets using DM techniques, namely pattern matching, clustering, rule association, regression, etc. The progressive growth of information technology has paved way to further explore Distributed/Collective Data Mining, Spatial and Geographic Data Mining, Temporal Data Mining, Spatio-Temporal Data Mining, Multimedia Data Mining and Phenomenal Data Mining. DM today performs computation on the database or warehouse at a single geographical location paving way for increased computation cost and questioning on data privacy. Future scope of DM is computing data located at different geographical locations. This is termed DDM/CDM (Collective Data Mining)[1].

The main factors which led to the evolution of DDM are – privacy of sensitive data, transmission cost, computation cost and memory cost. The objective of DDM is to extract useful information from data located at heterogeneous sites. Distributed computing comprises distributed sites, hosting computing units at individual heterogeneous points. DDM follows decentralized mining strategy which differs from centralized strategy making entire working system scalable by distributing workload across heterogeneous sites[1].

Alfredo Cuzzocrea[2] stated that framing a methodology for DDM is challenging not only by distributed environment, but also for its efficient resource sharing and minimized computational complexity specifications. DDM mainly comprises of two variations — data distributed and computation distributed. In the former method, data is distributed among heterogeneous sites at local level and computation is hosted at global level. In the latter method, computation is distributed among heterogeneous sites at local level and data is hosted at global level. Figure 1 explains DDM working architecture. The database of heterogeneous sites hosts useful, unknown information. DDM algorithms will be applied over data at heterogeneous sites as local model and finally the DM computed result will be agglomerated to form global model [1].

Kargupta et al.[3] and Zaki et al. [4] discussed that several researchers analyzed the complexity involved in framing methodology for DDM in two ways: analyzing on effective and efficient usage of computational resources at individual distributed data-sites, performing knowledge discovery at individual distributed data site (local level) and aggregating knowledge discovered at global level.

Fu Y et al.[5] discussed certain issues in developing DDM algorithms, namely formulating suitable DDM algorithms for heterogeneous datasets, minimizing computational and space complexity, enhancing data privacy at distributed sites and maintaining local datasets autonomy. Further, all these issues are interrelated to each other. This has set way to many researchers to carry-out their work in this field. Park et al. [6] developed an architecture for DDM where processing takes place locally at individual data sites. Finally data is accumulated to form global level. Grigorios Tsoumakas et al. [7] presented architecture for DDM where knowledge acquired from local distributed data sites is accumulated at global level forming a merger site.

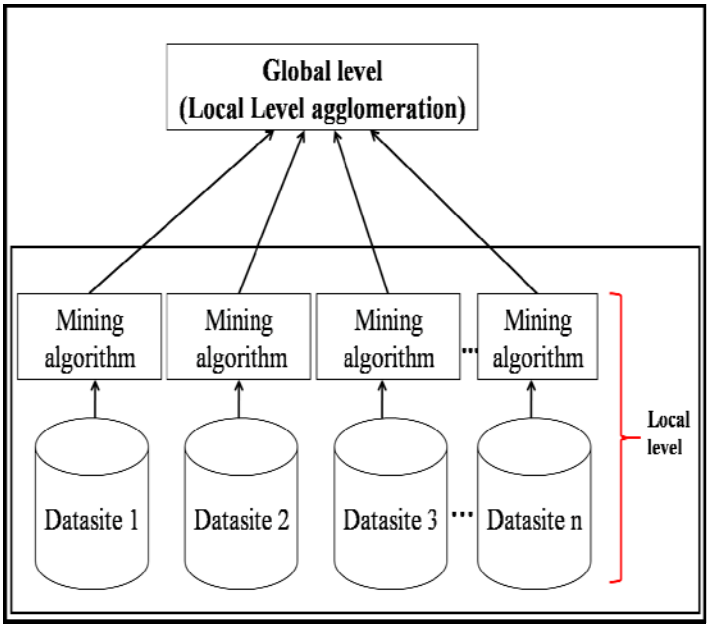


Fig. 1. Working Architecture – Distributed Data Mining

A. *DDM architectural models*

DDM comprises of two variations of architectural models: Client-Server based DDM architecture and Agent-based DDM architecture based on collection/processing of data /code at local or global levels[1].

Client-Server based DDM

Client-Server based DDM architecture is shown in Figure 2 where client sending a request to DDM server, which in-turn authorizes the targeted data to collect at local level. The accumulated data from heterogeneous sites (local level) has to be processed at global level. Client-Server DDM architecture has data migration complexity (transmitting all the data to perform mining at global level) and therefore this increases network bandwidth and network latency[1].

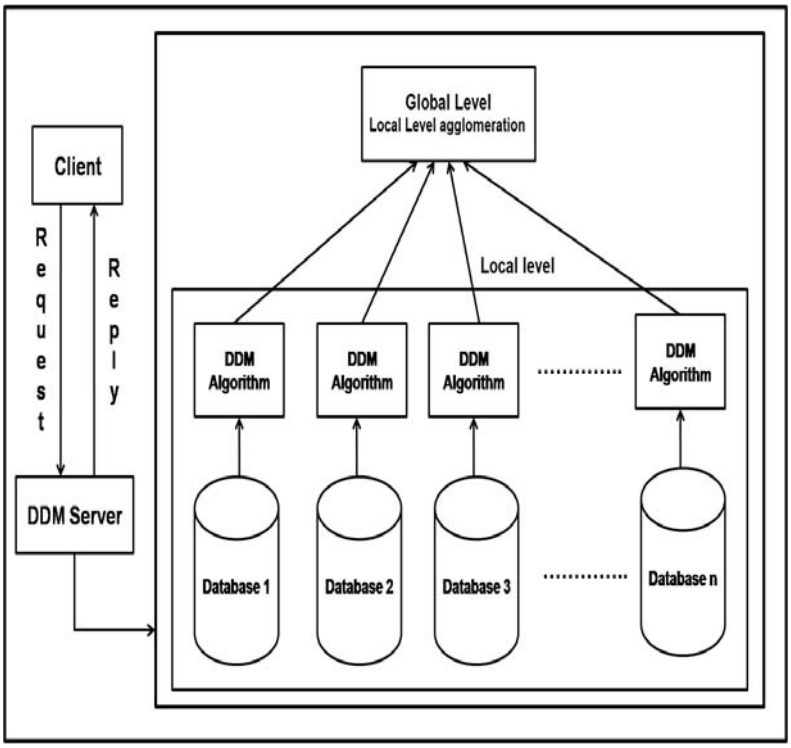


Fig. 2. Client-Server based DDM

DDM evolution along with its retrieval approaches is shown in Figure 3.

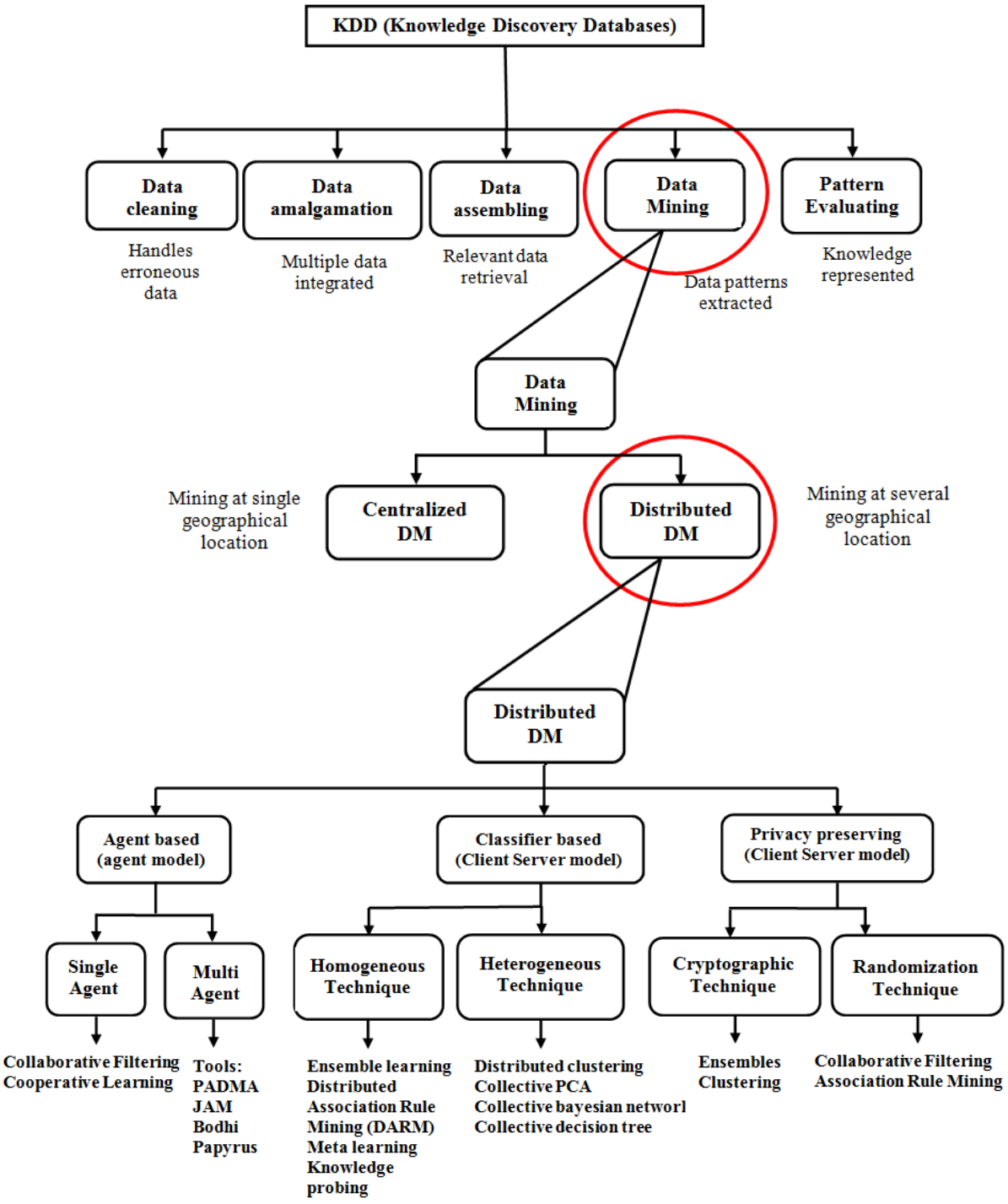


Fig.3. DDM classification

Agent based DDM (Agent model)

Agent-based DDM architecture is shown in Figure 4 where client generates multiple Mobile-Agent based Data Mining (MADM) agents for each data-server at heterogeneous sites at local level and the result will be sent back to the client. Further, certain knowledge integration approach takes place at client at global level to integrate results obtained from different MADMs.

Agent-based DDM architecture incorporates code movement instead of data, providing scalable and efficient approach. Limitations are inefficient use of computational resources and partial-involvement of DDM server[1].

Agent-based DDM architecture has two sub-variations, namely, stationary-based agent DDM architecture and mobile-based agent DDM architecture, both variations based on self-directed migrations by agents.

In the former case, agents are passive whereas in latter case agents are active.

These two models are used to develop various DDM approaches like classifier, agent and privacy based and are discussed along with their research contributions in the following sections[1].

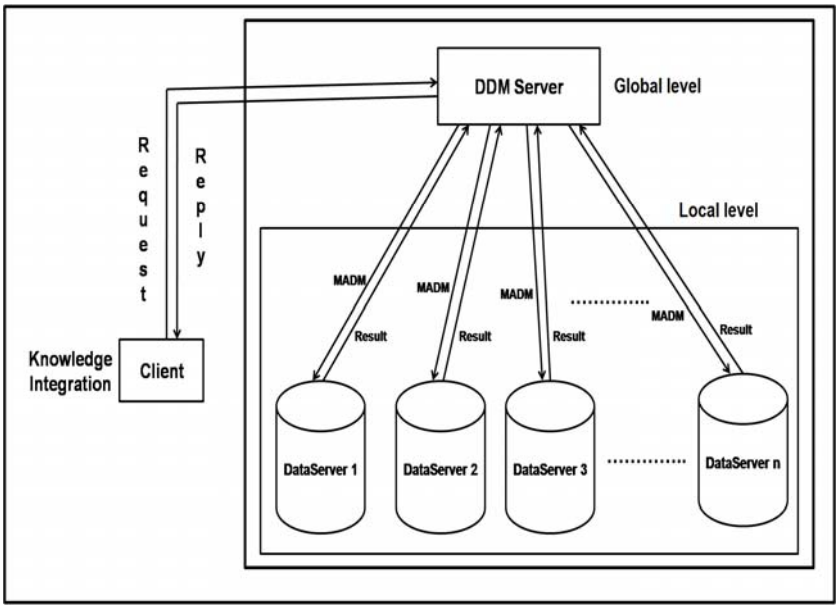


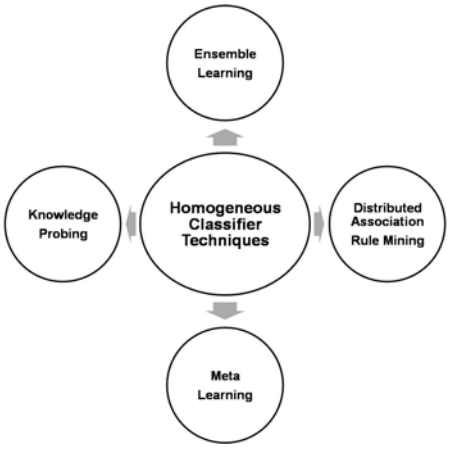
Fig. 4. Agent-based DDM

The organization of the paper is as follows: Section II describes an overview of DDM based on classifier approach and its related works. Section III discusses works on agent-based DDM and multi-agent systems. Section IV discusses works on privacy-preserving based DDM approaches. Section V concludes the paper.

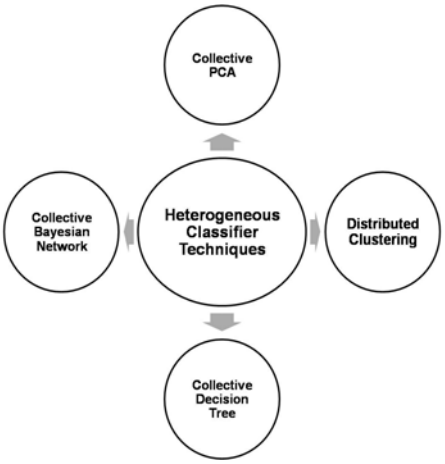
II. CLASSIFIER-BASED APPROACH FOR DDM

Classifier-based approach for DDM is applied to evaluate the pattern/association between distributed data. It is based on client-server model where data is accumulated and processed at global level.

Based on datasets considered, classifier approach of DDM can be classified into two approaches: homogeneous classifier (mining distributed data-sites involving similar attributes) and heterogeneous classifier (mining distributed data-sites involving distinct attributes) approaches.



(a)



(b)

Fig. 5. (a) Homogeneous Classifier DDM Techniques and (b) Heterogeneous Classifier DDM Techniques

DDM classifier approaches (homogeneous and heterogeneous classifier) along with their 4 techniques in each approaches is shown in Figure 5 and are surveyed as follows[8].

A. Homogeneous classifier techniques for DDM

Homogeneous classifier techniques for DDM deals with the mining of similar attribute data. Four techniques in this are ensemble learning, Distributed Association Rule Mining (DARM), meta-learning and knowledge-probing.

Ensemble Learning

An Ensemble Learning technique requires multiple learning models to obtain final predictions. An ensemble learning classifier approach proved to be an effective learning approach, in-terms of combining multiple learning models giving better prediction result than any of the solo classifier approach[8]. It is believed that an ensemble technique is better compared to any single classifier technique proposed by Breiman, bagging(1996), arcing(1998) and random forest(2001) [9]. Other techniques are boosting and stacking. Out of these five ensemble learning classifier approaches, bagging and boosting have been proved as an effective ensemble learning classifier techniques[10]. Though the above said five ensemble learning techniques takes maximum computation time and memory cost, the final classifier result is effective. It is suitable for a small-dataset, but when it comes to larger dataset it resulted in increased computation time.

Yan Li et al. [11] discussed a distributed ensemble technique for mining health care data under privacy constraints. Proposed a novel privacy-based distributed ensemble classifier technique, adaptive privacy-based boosting for predicting model for EHR data. By this technique each distributed site, had been able to learn data distribution effectively and share medical data without revealing sensitive patient data achieving less computational complexity and communication cost.

Distributed Association Rule Mining (DARM)

DARM incorporates certain association rules for generating local datasets. Finally, the global datasets is generated from multiple local datasets[10]. Vinaya Sawant et al.[1] compared three algorithms of DARM. Count Distribution algorithm includes apriori algorithm generating k-itemsets for each iteration at local level, global level computes the final-itemsets. The Fast DARM algorithm comprises the pruning of itemsets at local level where pruning is followed for each iteration. The Optimized DARM algorithm includes both Count Distribution algorithm and Fast DARM algorithm. It performs efficiently than former two algorithms by deleting earlier itemsets at local level and deleting duplicate transactions by keeping track of a counter[10].

Kawuu W.Lin et al. [12] discussed a fast and resource efficient mining algorithm for discovering frequent patterns in distributed computing environments. An automatic allocation of local-level nodes for detecting frequent patterns is discussed. Progressive Size Working Set (PSWS) method encompasses initial assignment of computing nodes for each transaction leading to decreased load-balancing effect. It dynamically decides the number of computing nodes needed. The proposed FP-growth (Frequent Pattern-growth) mining algorithm don't involve any parameter, but still able to discover patterns, without initially setting the required number of nodes leading to efficient load-balancing, execution and network transmission cost.

Ogunde et al. [13] discussed a partition enhanced mining algorithm for DARM systems. The proposed ARM agent is capable of assigning coordinating agents, which receives requests and determines the required number of geographical sites. The work includes 2-phase, wherein phase 1 horizontal segmentation of a dataset into smaller transactions occur. At phase 2 local iterated datasets are integrated globally to form global iterated datasets achieving dynamic decision on number of computing nodes.

Sunil Kumar et al.[14] discussed an apriori algorithm in distributed mining on XML (eXtendible Markup Language) data. The proposed algorithm ODAM (Optimal Association Rule Mining), mining process in parallel leading to better response time and minimized communication cost. ODAM removed infrequent transactions and place in main memory, reducing transaction size. If inserted transaction is already in memory counter is updated to +1 otherwise it inserts the transaction and updates the counter to 1.

Frank S.C. Tseng et al.[15] discussed DARM boosting by data de-clustering where datasets will be de-clustered into partitions. Round-robin method had been followed by iteratively assigning dataset to individual geographic datasites. Load-balancing approach had been followed where item-sets of each geographic site is generated quickly. Performance varied with size of cluster and items of cluster leading to decreased communication cost and space complexity.

Golam Kaosar et al.[16] discussed DARM with minimum communication overhead by incorporating message passing interface. Global frequent large item-sets were generated thereby reducing the communication overhead among neighboring nodes. It minimizes communication overhead by transmitting binary vector and frequently invoked datasets count. Message passing interface along with FDM (Fast Distributed Mining) pruning technique helped to reduce communication overhead across distributed geographical data sites and achieving decreased communication overhead.

Meta-Learning

Meta-learning classifier approach denotes use of meta-classifier and base-classifier. This classifier approach proved to be effective, scalable, portable, compatible, extensible and efficient. Meta-learning includes both arbitration and combining. Arbitration generates final prediction result of the feature vector. Combining generates final prediction based on classifier output and classification output or based on classifier output, classification output and feature-vector prediction[10].

Yihong Dong et al.[17] discussed clustering algorithm based on data partitioning for unevenly distributed datasets. The proposed new clustering algorithm has been used for uneven datasets. A Fuzzy Connectedness Graph algorithm (PFHC) had been developed based on partitioning uneven datasets into similar datasets with equal density. FHC is used to obtain clusters with equal density in the local level of distributed sites. Finally the integration of several local clusters to global clusters is done leading to mine uneven datasets efficiently.

Josenildo Costa da Silva et al.[18] discussed distributed data clustering inferences by proposal of kernel-based distributed clustering scheme (KDEC-S) algorithm. A helper site for each local level distributed site builds a local-level estimate and forward to peer distributed sites. Local-density clustering algorithm is finally built with global-level estimate achieving data confidentiality.

Lamine M.Aouad et al.[19] discussed lightweight clustering technique which minimized communication overhead by minimum variance formation of clusters.

Knowledge-Probing

Knowledge-probing is defined as combining several local models to generate the final global model. Steps involved in knowledge-probing include generating base-classifier from an off-the-shelf classifier model, selecting untagged data for probe set, preparing probe set by accumulating final result from base-classifier and finally generating final prediction model of the probe data set.

The main difference between knowledge-probing and meta-learning is: knowledge-probing is relying on probe data set for its final prediction, whereas meta-learning involves arbitration and combining learning methods for the final prediction[10].

Yike Guo et al.[20] discussed knowledge probing in DDM where two-stage process takes place. A collection of base-classifiers is trained in first stage. In second-stage meta-learning technique is applied to attributes for prediction. Further, collaboration of knowledge-probing and other traditional DM algorithms resulted in improved performance.

B. Heterogeneous classifier techniques for DDM

Heterogeneous classifier techniques for DDM deals with the mining of distinct attribute data. Four techniques in this are collective principal component analysis, distributed clustering, collective decision tree and collective bayesian learning.

Collective principle component analysis

Principle component analysis (PCA) is used for predictive models, done by factorizing based on eigen vector and eigen values. Collective PCA performs PCA on local datasets, by selected eigen vector set. Global dataset prediction result is obtained by, combining selected dominant eigen vector sets obtained by PCA on local datasets thereby classifying the dataset by heterogeneous approach[10].

Zheng Jian Bai et al.[21] discussed PCA based clustering algorithm by which truncation of singular value decomposition (SVD) reduces communication costs leading to error reduction and effective for smaller data sets.

Distributed clustering

CHC algorithm inculpates dendrogram, a tree representation of clusters. Local dendrograms is generated at each local geographical site. Global dendrogram is generated from multiple transmitted local dendrograms. RACHET, Hierarchical clustering algorithm is generated at each local geographical site; separate statistics set is generated for each site. Global level agglomerates local dendrogram to generate final predictions. DBDC algorithm incriminates generating a local cluster prediction model, at each heterogeneous local site. Representative points of each cluster set is selected and finally they were combined at global level for the final prediction[10].

Josenildo Costa da Silva et al.[18] discussed distributed data clustering inferences by proposal of kernel-based distributed clustering scheme (KDEC-S) algorithm. A helper site for each local level distributed site builds a local-level estimate and forward to peer distributed sites. Local-density clustering algorithm is finally built with global-level estimate achieving data confidentiality.

Trilok Nath Pandey et al.[21] discussed mobile-agent based distributed mining dealing with query optimization, discovery plan, local knowledge discovery and knowledge consolidation by distributed clustering.

Collective decision tree

Collective decision tree implicates decision tree generation at the local geographical heterogeneous site. Global level prediction result is accumulation of local decision trees thereby classifying the dataset by heterogeneous approach[10].

Sung Baik et al.[23] discussed combining decision tree and agent approach for network intrusion detection. Classification rules learned by decision trees were used for detecting network intrusion. Agents working collaboratively, collect details of network intrusion which partially update results in the form of indices of records.

Collective bayesian learning

Collective bayesian learning enfolds bayesian learning model generation at the local geographical heterogeneous site. Global level prediction result is an accumulation of local bayesian learning models thereby classifying the dataset by heterogeneous approach[10]. Kamalika Das et al.[24] discussed two algorithms L-ring (Local Ring) and PAFS (Peer-Peer Feature Selection) in an asynchronous manner. Each peer decides its own privacy constraints. The local interaction among participating nodes resulting in minimized communication complexity.

III. AGENT-BASED APPROACH FOR DDM

Agent technology in DDM seems to be enhancing efficiency and scalability by reducing network traffic (bandwidth reduced by sharing only the code and not the data among distributed sites), allows the system to scale efficiently without increasing the complexity of the system (since an agent can spawn another agent), agents react dynamically and adapt easily to the changing environment, agents operate asynchronously by which agents disconnect from the network and reconnect automatically after its tasks and agents are fault-tolerant by which it bypasses a fault distributed data site and shelters on a reliable distributed data site[1].

Multi-Agent Systems (MAS), a technology by which multiple agents coordinate together and perform mining process, particularly suitable for DDM by which computation time and memory cost can be reduced by maintaining a pool of agents in global-level and deploying multiple agents to each distributed data sites. Agents further are capable of adapting to errors and faults which arise in the entire system by interacting with all the distributed data sites. Multi-Agent Systems (MAS) comprises multiple agents capable of achieving tasks difficult to be achieved by a single agent. MASs an artificial intelligence technology, success lies in the way of coordinating individual agent behaviors[1].

Xavier Lim´on et al.[25] discussed an agent and artifact approach to DDM by Collaborative Filtering (CF) among agents. Decision tree technique is used which generated decision tree among each distributed data sites. Global level final prediction is based on accumulated local-decision trees. CF strategy had been implemented with J48 decision tree. MAS were composed of coordinating agent and workers.

Mainly 3 artifacts were used by agents, namely J48 classifier, InstanceBase and Oracle. Coordinating agents utilize Oracle for extracting information for learning data and worker agents use the split data for processing. Agents store the data in InstanceBase. An initial CF strategy model is built by J48 classifier. Mainly Adult, German, letter, poker and waveform datasets from UCI (University of California, Irvine) repository were used.

Jie Gao et al.[26] discussed enhancing DDM by a framework called CoLe^2, agent-based DDM mining model. It is termed as Cole^2 because it comprises two-loops working cooperatively. The two loops operating are inner-cooperative and outer-adjustment loops. There were three agents operating cooperatively. Cooperative loop allowed for knowledge-based strategies used by controller agent. Adjustment loop helped in coordinating all agents by knowledge-base. Controller agent controlled the entire system. Miner agents were core of the mining process in this technique.

Heterogeneous miners agents were used which utilize several mining algorithms. Combiners' agents were employed in creating rules by combining the types. Further, combiner agents assigns miner for mining and combining to a set of rules. Two medical datasets were implemented with CoLe^2, one of diabetes and other kidney disease dataset. The running time for kidney dataset is 65.79% and the running time for diabetes dataset is 18.39% therby achieving efficiency and comparable result quality.

Vladimir Gorodetsy et al.[27] discussed agents and DDM in Smart Space environment particularly for overlay network. Choice of virtual neighbors is done in various modes. A peer-peer agent-based technique is followed which facilitated flexibility issue for easily adapting to human needs. Context-driven DM approach is utilized with basic smart space environment. This approach is implemented only for human needs and utilizes decision making capability, suitable only for situational awareness and not for intentional awareness.

Matthias Klusch et al.[28] discussed collaborative DDM, multi-strategy DDM. Further KDEC (Kernel-Distributed clustering based on density estimation) is implemented by a density estimation based on probability values. KDEC scheme is familiar for 3 issues, namely density estimates, de-clustering computation and compact dataset representation for transmission purpose.

G.S. Bhamra et al.[29] discussed agent enriched DARM, combining agents and ARM. An agent enriched DARM framework technique is proposed, AeMSAR (Agent enriched Mining of Strong Association Rules) is discussed. This framework included a central store, where accumulation of local knowledge is stored and heterogeneous sites included transactional datasets. Central site is responsible for dispatching agents which carried information to and fro the distributed sites. Both mobile and stationary agents were stored in the agent pool. Three mobile agents and 2 stationary agents were utilized from the agent pool for performing different tasks. The 3 mobile agents were responsible for result transportation and other for storing state variables. The 2 stationary agents were responsible for global knowledge maintenance and keeping track of frequent item-set generator.

Yue Fuqiang et al.[30] discussed distributed data stream mining with mobile agents. Mobile agent mines data streams for solving noise present in data processing, slower classification and inefficient mining problem. This model includes efficient mining, reduction of erroneous data and inefficient data pre-processing. Mobile agent-based distributed data stream processing model (DDMMMA) is proposed which computes the number of resources; divides tasks into sub-tasks for deciding on which agent to reach which host node, etc. Mobile-agent mechanism for migration is proposed for shortest path algorithm for performing the tasks.

Vuda Sreenivasa Rao et al.[31] discussed a novel framework by interaction and integration among agent and DM for dimensions, namely learning, knowledge, interface, interaction, social, application and performance. Each layer worked collaboratively with one another.

Chayapol Moemeng et al.[32] discussed a new agent based DDM model called, i-analyst by management of resources phase and execution phase. First phase is responsible for project management, DM model, instance management, interaction with users with set of modules for algorithm management and user-access-privilege management. Second phase is responsible for maintaining a resource database of datasets, system resources, DM models, etc. The central container hosted persistent agent called DSA (Service Agent). Activity runners have performed DM and analysis tasks. 3 Application Programming Interfaces were used for communication among the two-phases.

Xining Li et al.[33] discussed deploying mobile agents for searching data which were distributed. It wandered around and generated global datasets from local datasets. It is able to overcome network congestion, security and unreliability. Further, focusing on efficiency leads to database connection management to reduce network cost reusing network connections. A cache management system which enabled to store datasets from the database and to restore frequent items in local-level of distributed sites. Further, delayed evaluation prevented redundant information. Establishment, maintenance and termination of database connections were the major problems. Agents shared a number of pre-defined database connections, but unnecessary termination and establishment of database connections must be avoided.

U.P. Kulkarni et al.[34] discussed mobile agent based DDM by which knowledge from distributed sites is extracted by association rules. The main objective of this approach is to reduce the time required for computation of global-frequent item-set. The proposed algorithm comprises of two-phase: local-level distributed sites sends local frequent item-set to a central site and also to its neighbor local-level sites. Calculation of item-set at global level is done in an overlapped fashion in local site. Total time for communication is reduced; local-level site doesn't need to wait for global site to send item-sets which were frequent.

Josenildo C. da Silva et al.[18] surveyed on multi-agent based DDM approaches. Further, privacy-preserving algorithms and distributed clustering algorithms were discussed. A new technique is framed which concentrated on essential approximation rather than density estimation. The algorithm had two parts: helper and local peer. Local peer is density-based, general approach to clustering. The helper had sampled points from all peers. Attackers were not able to reconstruct the data since the information is not stored in the kernel, helper maintained the stored data.

Sung Baik et al.[24] discussed agent-based decision tree algorithm by which communication cost and knowledge integration costs were minimized. It is performing better than centralized decision tree algorithm since the entropy calculation is critical for huge datasets. Decision rule generated at each agent and it is notified to the mediator for termination. Processing time decreased with an increase in the number of agents involved.

A. Multi-Agent Systems

Some of the MAS (Multi-Agent System) of DDM are PADMA (Parallel DM Agents), JAM (Java Agents for Meta-learning), bodhi and papyrus. The architecture of above MASs with their components and interface between them are discussed in the following sub-sections.

PADMA: Parallel Data Mining Agents

PADMA a multi-agent based model for parallel DM. Parallel Data Mining Agents (PADMA) implements text classification for parallel systems. It includes 3 functions such as accessing data in-parallel, clustering hierarchically and data visualization. PADMA is developed not for a specific domain; it is implemented for unstructured document classification. As shown in Figure 6 PADMA activity diagram comprises of 3 main modules: DM agent, facilitator as coordinator for the agents and a user interface for result visualization. Each DM agent specialized in unstructured document classification extracts higher-level information from data and forward to facilitator. Agents working in parallel forward the collected information to the facilitator.

Facilitator acting as agent coordinator provides the data to user. Facilitator-user communication is based on SQL queries. PADMA is implemented on cluster of sun sparc workstations and on IBM SP2. It is portable to any distributed machine. Agents in PADMA provides parallel mining in relational database useful for exploiting the section for text classification by clustering [35].

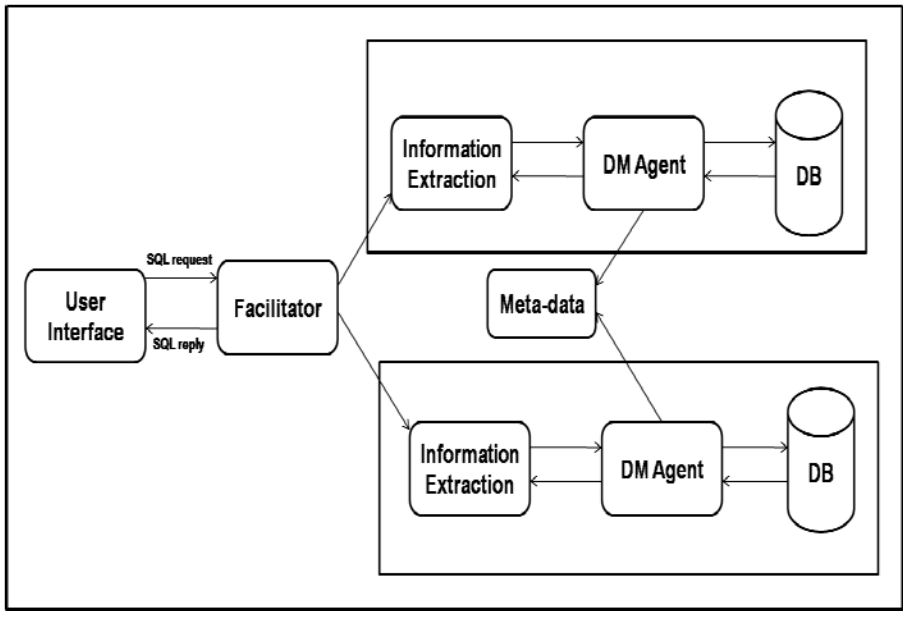


Fig..6. UML activity - PADMA based on hierarchical clustering

Since SQL operations are supported by PADMA, queries namely parallel select and join operations is used. It further supports 5 operations, namely create, delete, read, query and clustering. Each individual agent carries select and join operations on its local data and the facilitator collects data from each agent without any inter-process communication among agents.

Three join operations are used, namely, nested join, sort-merge join and hash join. Nested join takes place by comparing rows of both the tables.

Sort-merge join takes place by binary search comparison of sorted table by attribute values. Hash-join takes place by partitioning sorted tables based on attribute values into several buckets. Each table results from the agent are forwarded to facilitator which compares with another table result of another agent.

PADMA is implemented with hierarchical clustering (unstructured document) in which each hierarchy level includes concepts (attributes). The user interface provides visual interaction with the system. Hierarchical clustering provides an interactive clicking by which top-level cluster hierarchy is presented to the user initially.

Then the user clicks on a cluster, which further explores next hierarchy levels. It depicts an improved response time and computation time. [35].

JAM: Java Agents for Meta-learning

Java Agents for Meta-Learning (JAM) implements DM application for meta-learning. It implements distributed meta-learning technique and classification techniques linked through a number of data-sites. JAM with modules is shown in Figure. 7.

Each agent at local level computes local data-site classifiers on local database. Each data-site computes final classifier from data obtained from its peer data-sites using local meta-learning agent. At each data-site, datasets are classified and labeled independently.

Through a configuration file at each data-site, number of attributes to be classified is noted to the system by user. The user interface of JAM administers the meta-learning approach and dynamically facilitated agent exchanges.

JAM is implemented with 3 data-sites Marmalade, Mango and Strawberry. JAM depicted improved computation time[36].

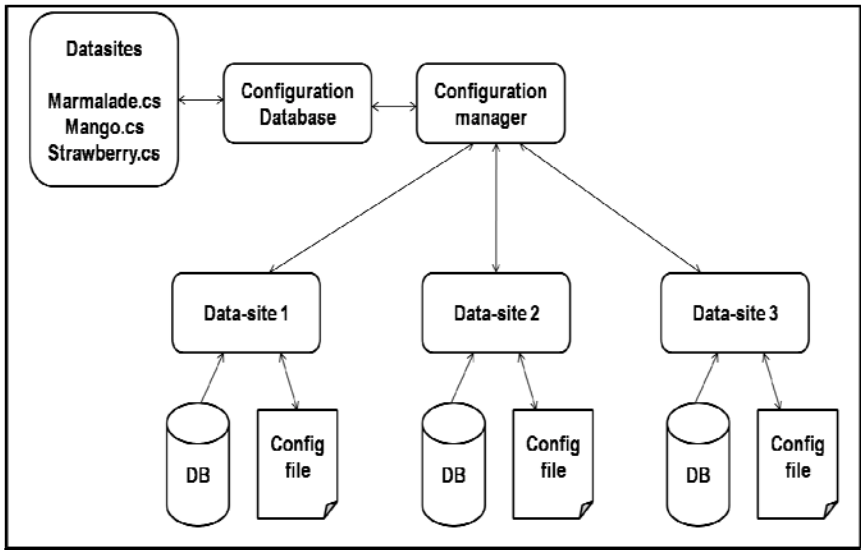


Fig. 7. UML activity - JAM architecture based on meta-learning

Bodhi

Bodhi an agent and Java based DDM system. Bodhi is capable of transferring agents along with its configuration, state, environment and knowledge learned from a distributed location to another distributed location. Bodhi includes mobile agents, agent station and facilitator along with user interface for communicating with the system. Bodhi activity diagram is shown in Figure 8[36].

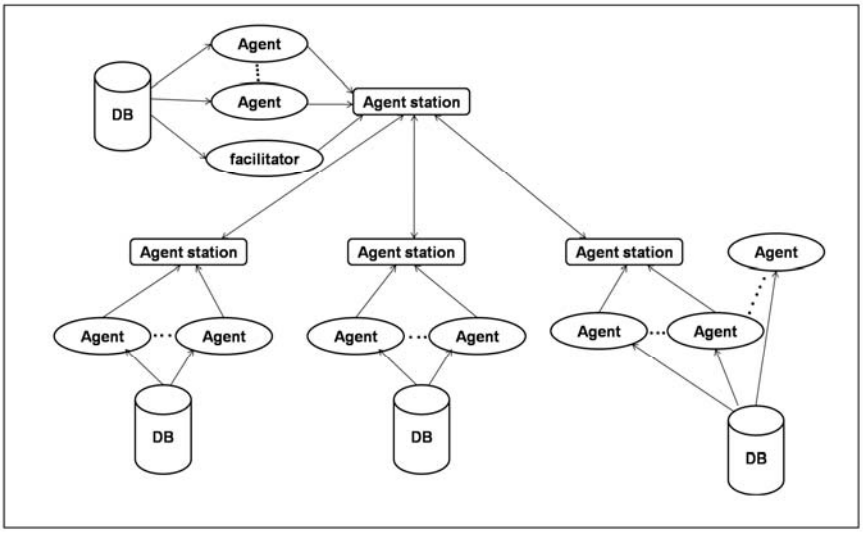


Fig. 8.UML activity – Bodhi based on meta-learning

Papyrus

Papyrus, a layered system consists of DDMs services. It consists of four layers designated access tools and network services. Papyrus layered-architecture is shown in Figure 9. The first level is called *Osiris* (data management layer) supports meta and super clusters of DM. Further, Osiris layer divides data into smaller individual parts called *folios*, further divided as *segments*. Thus, it is possible to move segments among clusters. The second level is called *DM layer* which includes extraction of learning set from data and inclusive of DM algorithm for semi-automatic production of the predictive rule set. The final output is PMML (Predictive Model Markup Language) files. The third layer is called *predictive modeling layer* which supports and manages predictive models productions. The uppermost (fourth) layer is called an *agent layer* (bast) which identifies relevant clusters within meta-clusters and super-clusters[37].

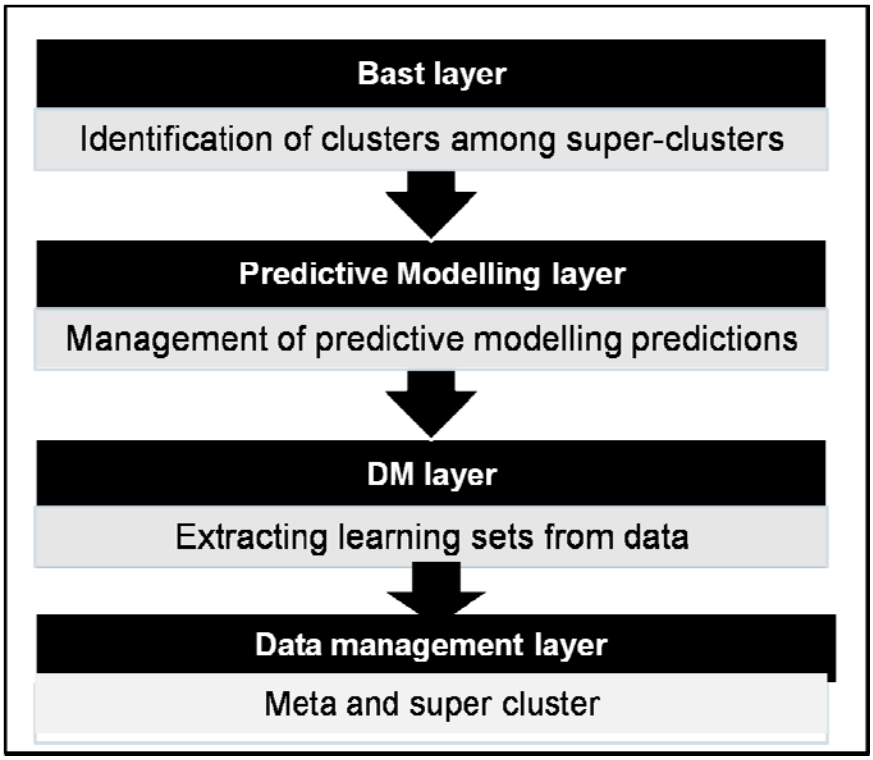


Fig.9. Layered architecture - Papyrus based on clustering

IV. PRIVACY-PRESERVING BASED APPROACH FOR DDM

Enormous amount of personal data are collected regularly and mined for information. Data includes patients' medical history, credit card transactions, military records, etc. On mining those data for obtaining patterns/association threatens privacy of original data. Privacy Preserving DM (PPDM) aims to protect data from unauthorized exposure. It becomes an issue when different data owners' wans to access the knowledge from the data by several frequent pattern techniques, but they inclusively reveal their data at computation phase. Privacy-preserving DDM success relies on building a valid DDM model for finding useful data associations but hiding the data from others. Mainly privacy preserving DDM model is built with classification, clustering and ARM[38].

It is based on client-server model where data is accumulated and processed at global level. Privacy preserving DDM is implemented in two-ways: adopting cryptographic techniques for providing secure transactions in distributed model or adopting randomization techniques by randomizing original data. Though randomization technique is a better approach it suffers from accuracy when privacy is at its peak, but cryptographic technique provided better accuracy and privacy than randomization technique. Privacy preserving DDM is particularly applicable in almost all mining areas, namely clustering, ARM, bayesian model, decision tree, ensemble methods and CF[38].

In recent years privacy preserving medical record mining is on demand since the need to preserve privacy of patients' medical record is on the rise. Yan Li et al.[11] discussed a distributed ensemble technique for mining health care data under privacy constraints. Proposed a novel privacy-based distributed ensemble classifier technique, adaptive privacy-based boosting for predicting model for EHR data. By this technique each distributed site, have been able to learn data distribution effectively and share medical data without revealing sensitive patient data achieving less computational complexity and communication cost.

Masooda Modak, et al.[38] discussed secured ARM on partitioned data. Vertically partitioned data used distributed Apriori T-tree algorithm along with vertical partitioning. For horizontally partitioned data, collaborative approach is followed by which only the calculated global association rules value is revealed to its horizontal parties. From the global association rules generated each distributed site decides on whether to publish its own global rules to the other distributed site or not.

There is a controller taking responsibility of performing computations for all the distributed data sites. Based on hierarchy concept, association rules generated at each distributed data site is hidden along with the sensitive information.

Yiannis Kokkinos et al.[39] discussed DDM privacy-preserving for ensemble technique. Neural classification is selected by confidence ratio affinity propagation by privacy computing. Ensemble classifiers classify local-level data. For validation, training sets were used. Confidence ratio affinity between each two site finds the most suitable confidence ratios. The method calculates confidence ratio affinity propagation among classifiers and final pruning is done. It is implemented on the UCI and KEEL (Knowledge Extraction based on Evolutionary Learning) data repository show reduced time complexity of O(CE) for C classifiers and E examples.

Hemanta Kumar Bhuyan et al.[40] discussed privacy preserving sub-feature selection in DDM by fuzzy method, maintain the privacy of original data. Two-fuzzy sets were generated using borel set, which helped in determining sub-feature selection within a certain interval.

Sub-feature selection by fuzzy method showed effective and better performance compared to traditional methods. Privacy of original data is maintained. Experimental implementation of hepatitis (157 instances), yeast (1484 instances) and heart disease datasets revealed reduced computation time than conventional approaches achieving efficient sub-feature selection and privacy of original data.

Feng Zhang et al.[41] discussed privacy-preserving two-party DARM on horizontally partitioned data by encryption, done among peers for secure division. Experimental results revealed increased computation cost and communication cost than conventional approaches.

Zhuojia Xu et al.[42] discussed privacy preserving DDM by four dimensions, namely data partitioning model, mining algorithm, secure communication model and privacy preservation techniques. Data partitioning model included either homogeneous or heterogeneous approaches. DDM algorithm comprises ensembles, ARM and clustering. Communications were secured by either multi-party or third-party operations. Cryptographic techniques included privacy preserving techniques, namely public-key encryption, secret transfer, RSA algorithm, etc. This classification presented to be a guiding light for further progressing in privacy-preserving technique of DDM in forthcoming years.

Xinjun Qi et al.[43] discussed DDM privacy preserving classification by data distribution, data distortion and DDM techniques. Further, it discussed several privacy protection technologies, namely, ARM by perturbation on hiding association rules. Only association rule values had been revealed to neighboring distributed data sites.

Yanguang Shen et al.[44] discussed personalized privacy preserving DDM which combined multi-party secure computations and K-anonymity technique along with decision tree classification. Initially non-sensitive data is anonymized which is useful for classifying the data, but becomes useless for sharing with other distributed data sites. Finally, sensitive data is shared via multi-party secure computation technique to prevent sensitive data leakage. From the datasets considered two-divisions of sensitive data and non-sensitive data were computed. Sensitive data were shared via multi-party secure computations and non-sensitive data are shared via K-anonymity technique. K-anonymity technique implemented the generalization concept by which linking rate is reduced by cutting connections of sensitive attribute, protecting the privacy of sensitive information. Information gain for each attribute is computed collaboratively building decision tree and choosing attribute with maximum value as the targeted information. If the attribute targeted is sensitive then it has to recover from multi-party secure computations achieving efficiency and minimum overhead in communication and cost.

Rebecca N.Wright et al.[45] discussed bayesian network-privacy preserving for DDM. DDM privacy preserving based on bayesian network divides the problem into smaller sub-problems for efficient privacy-preserving concept. A secured two-party computation is applied on bayesian network. Experimental results were implemented which depicted efficiency and accuracy performance. Further, privacy preserving learning classification model fully distributed k-anonymization and anonymity preserving data collection is represented.

Xun Yi et al.[46] discussed privacy preserving DARM model by semi-trusted mixer. Data from the distributed data site is dispatched to the model which mixed the information and dispatched it to the other distributed data sites. A strong global association rule is obtained from the combination of distributed data sites. For a single data communication, only 2 messages (information) need to be sent and received to and fro the semi-trusted mixer and distributed data site achieving minimized communication cost and storage cost.

V. CONCLUSION

This paper has discussed the various approaches and techniques of DDM. Several recent research works presented in literature encompassing DDM approaches at global level or at local level have been reviewed for different applications. In the first and third approaches, works have been focusing on classifier-based DDM and privacy-preserving based DDM which are following client-server model. In these approaches, data is accumulated and processed at global level leading to decreased computational complexity, improved accuracy and efficiency. In the second approach, works have been focusing on agent-based DDM which are following agent model. In these approaches, data is accumulated and processed at local level leading to decreased space complexity and computation time. The authors expect that this research work could be a guiding light for proposing new techniques in the vast and upcoming field of DDM.

REFERENCES

[1] Vinaya Sawant and Ketan Shah, “A review of Distributed Data Mining using agents”, International Journal of Advanced Technology & Engineering Research (IJATER), vol. 3, no. 5, pp. 27-33,2013.
[2] Alfredo Cuzzocrea, “Models and algorithms for high-performance distributed data mining”, Elsevier Journal of Parallel and Distributed computing, vol. 73, no. 93, pp. 281-283, 2013.
[3] Kargupt., Kamath, and Chan, “Distributed and Parallel Data Mining: Emergence, Growth and Future Directions”, Advances in Distributed Data Mining, (eds.), Hillol Kargupta and Philip Chan, AAAI Press, pp. 407-416, 1999.
[4] Zaki M J and Pan Y, “Introduction: Recent Developments in Parallel and Distributed Data Mining”, Springer Journal of Distributed and Parallel Databases, vol. 11, no. 2, pp.123-127, 2002.
[5] Park B H, and Kargupta H, “Distributed Data Mining: Algorithms, Systems, and Applications”, In. Data mining handbook, 2002.
[6] Tsoumakas G, and Vlahavas I, “Distributed Data Mining”, Encyclopedia of Data Warehousing and Mining, 2nd Edition John Wang (Ed.), Idea Group Reference, pp. 709-715, 2008.
[7] Fu Y, “Distributed Data Mining: An Overview”, In: Newsletter of the IEEE Technical Committee on Distributed Processing, pp. 5–9, March, 2001.
[8] Hillol Kargupta, “An Introduction to Distributed Data Mining”, http://www.eecs.wsu.edu/~hillol
[9] Breiman, L, “Pasting small votes for classification in large databases and on-line. Machine Learning”, vol.36, pp.85-103.
[10] S. V. S. Ganga Devi, “A Survey On Distributed Data Mining And Its Trends”, International Journal of Research in Engineering & Technology (IJRET), Volume 2, Issue 3, March 2014, pp. 107-120.
[11] YanLi, ChangxinBai and ChandanK.Reddy, “A distributed ensemble approach for mining health care data under privacy constraints”, Journal of Information Sciences, volume 330, February 2016, pp. 245-259.
[12] Kawuu W. Lin, Sheng-Hao Chung, “ A fast and resource efficient mining algorithm for discovering Frequent patterns in distributed computing environments”, Journal of Future Generation Computer Systems, Volume 52, November 2015, pp. 49-58.
[13] A.O. Ogunde, O. Folorunso, A.S. Sodiya, “A partition enhanced mining algorithm for distributed association rule mining systems” Egyptian Informatics Journal, Volume 16, Issue 3, November 2015, pp. 297-307.
[14] Dr. C.Sunil Kumar, P.N.Santosh Kumar & Dr. C.Venugopal, “An Apriori Algorithm in Distributed Data Mining System”, Global Journal of Computer Science and Technology Software & Data Engineering, Volume 13, Issue 12, 2013.
[15] Md. Golam Kaosar, Zhuojia Xu, Xun Yi, “Distributed Association Rule Mining with Minimum Communication Overhead”, Proc. of the 8th Australasian Data Mining Conference (AusDM'09), Volume 101, pp. 17-23.

[16] Frank S.C. Tseng, Yen-Hung Kuo, Yueh-Min Huang, "Toward boosting distributed association rule mining by data de-clustering", Journal of Information Sciences, Volume 180, Issue 22, November 2010, pp. 4263-4289.
[17] Yihong Dong, ShaokaCao, KenChen, MaoshunHe, XiaoyingTai, "PFHC: A clustering algorithm based on data partitioning for unevenly distributed datasets", Fuzzy sets and systems, Volume 160, 2009, pp. 1886-1901.
[18] Josenildo Costa da Silva, Matthias Klusch, " Inferences in Distributed Data Mining", Engineering Applications of Artificial Intelligence, Volume 19, 2006, pp. 363–369.
[19] Lamine M. Aouad, Nhien-An Le-Khac, and Tahar M. Kechadi, Lightweight Clustering Technique for Distributed Data Mining Applications", ICDM 2007, pp. 120–134, 2007.
[20] Yike Guo and Janjao Sutiwaraphun, "Probing Knowledge in Distributed Data Mining", PAKDD'99, pp. 443–452, 1999.
[21] Zheng-Jian Bai, Raymond H. Chan, Franklin T. Luk, "Principal Component Analysis for Distributed Data Sets with Updating", Advanced Parallel Processing Technologies, Volume 3756, 2005, pp. 471-483.
[22] Trilok Nath Pandey, Niranjan Panda, Pravat Kumar Sahu, "Improving performance of distributed data mining (DDM) with multi-agent system", (IJCSI) International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3, March 2012, pp. 74-82.
[23] Kamalika Das, Kanishka Bhaduri, Hillol Kargupta, "A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to peer networks", Journal of Knowledge and Information Systems, Volume 24, Issue 3, September 2010, pp. 341-367.
[24] Sung Baik, Jerzy Bala, "A Decision Tree Algorithm for Distributed Data Mining: Towards Network Intrusion Detection", Computational Science and Its Applications – ICCSA 2004, Volume 3046 of the series Lecture Notes in Computer Science, pp 206-212.
[25] Xavier Lim´on, Alejandro Guerra-Hern´andez, Nicandro Cruz-Ram´ırez, Francisco Grimaldo, "An Agents and Artifacts Approach to Distributed Data Mining", Advances in Soft Computing and Its Applications, Volume 8266, 2013, pp. 338-349.
[26] Jie Gao and J¨org Denzinger , "Improving the Efficiency of Distributed Data Mining Using an Adjustment Work Flow", Machine Learning and Data Mining in Pattern Recognition, Volume 7988, 2013, pp. 69-83.
[27] Vladimir Gorodetsy, "Agents and Distributed Data Mining in SmartSpace: Challenges and Perspectives", ADMI 2012, pp. 153–165, 2013.
[28] Matthias Klusch, Stefano Lodi, Gianluca Moro, "The Role of Agents in Distributed Data Mining: Issues and Benefits" Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03), 2003.
[29] G.S Bhamra A.K. Verma, R.B. Patel , " Agent Enriched Distributed Association Rules Mining", ADMI 2011, pp. 30–45, 2012.
[30] Yue Fuqiang, "The Research on Distributed Data Stream Mining based on Mobile Agent", Procedia Engineering , Volume 23, 2011, pp. 103 – 108.
[31] Vuda Sreenivasa Rao, S Vidyavathi, G.Ramaswamy, "Distributed Data Mining and Agent Mining Interaction and Integration: A Novel Approach", IJRRAS Volume 4, Issue 4, September 2010, pp. 388-398.
[32] Chayapol Moemeng, Xinhua Zhu, Longbing Cao, Chen Jiahang, "i-Analyst : An Agent-Based Distributed Data Mining Platform" , 2010 IEEE International Conference on Data Mining Workshops, pp. 1404-1406.
[33] Xining Li, JingBo Ni, "Deploying Mobile Agents in Distributed Data Mining", PAKDD 2007 Workshops, pp. 322–331, 2007.
[34] U.P.Kulkarni, K.K.Tangod, S. R. Mangalwede, A.R.Yardi, "Mobile Agent Based Distributed Data Mining", 10th International Database Engineering and Applications Symposium (IDEAS'06), 2006.
[35] Hillol Kargupta, Ilker Hamzaoglu, Brian Stafford, Vijay Hanagandi and Kevin Buescher, "PADMA: PArallel Data Mining Agents for Scalable Text Classification", High Performance Computing, 1991.
[36] Salvatore Stolfo, Andreas L. Prodromidis, Shelley Tselepis, Wenke Lee, Dave W. Fan, "JAM: Java Agents for Meta-Learning over Distributed Databases", KDD-97 Proceedings, 1997.
[37] S. Bailey, R Grossman, H. Sivakumar, and A. Turinsky, "Papyrus: A System for Data Mining over Local and Wide Area Clusters and Super-Clusters".
[38] Masooda Modak, Rizwana Shaikh, "Privacy Preserving Distributed Association Rule Hiding UsingConcept Hierarchy", 7th International Conference on Communication, Computing and Virtualization, Volume 29, 2016, pp. 993 – 1000.
[39] Yiannis Kokkinos, Konstantinos G.Margaritis, "Confidence ratio affinity propagate on in ensemble selection of neural network classifiers for distributed privacy-preserving data mining", Neuro-computing, Volume 150, 2015, pp. 513–528.
[40] Hemanta Kumar Bhuyan, Narendra Kumar Kamila, "Privacy preserving sub-feature selection in distributed data mining", Journal of Applied Soft Computing, Volume 36, November 2015, pp. 552-569.
[41] Feng Zhang, Chunming Rong, Gansen Zhao, Jinxia Wu, and Xiangning Wu, "Privacy-Preserving Two-Party Distributed Association Rules Mining on HorizontallyPartitioned Data", International Conference on Cloud Computing and Big Data, 2013, pp. 633-640.
[42] Zhuojia Xu, Xun Yi, "Classification of Privacy-preserving Distributed Data Mining Protocols", IEEE 2011, pp. 337-342.
[43] Xinjun Qi , Mingkui Zong, "An Overview of Privacy Preserving Data Mining", Procedia Environmental Sciences, Volume 12, 2012, pp. 1341 – 1347.
[44] Yanguang Shen, Hui Shao, Yan Li, "Research on the Personalized Privacy Preserving Distributed Data Mining", Second International Conference on Future Information Technology and Management Engineering, 2009, pp. 436-439.
[45] Rebecca N. Wright, Zhiqiang Yang, and Sheng Zhong, "Distributed Data Mining Protocols for Privacy: A Review of Some Recent Results", MADNES 2005, 2006, pp. 67–79.
[46] Xun Yi, Yanchun Zhang, 'Privacy-preserving distributed association rule mining via semi-trusted mixer", Data & Knowledge Engineering, Volume 63, 2007, pp. 550–567.