A Comprehensive Survey on Cloud Data Mining (CDM) Frameworks and Algorithms

HRISHAV BAKUL BARUA, *ACM Member*, Embedded Systems and Robotics Research group, TCS Research and Innovation Lab, Kolkata, India
KARTICK CHANDRA MONDAL, *ACM Member*, Department of Information Technology, Jadavpur University, Kolkata, India

Data mining is used for finding meaningful information out of a vast expanse of data. With the advent of Big Data concept, data mining has come to much more prominence. Discovering knowledge out of a gigantic volume of data efficiently is a major concern as the resources are limited. Cloud computing plays a major role in such a situation. Cloud data mining fuses the applicability of classical data mining with the promises of cloud computing. This allows it to perform knowledge discovery out of huge volumes of data with efficiency. This article presents the existing frameworks, services, platforms, and algorithms for cloud data mining. The frameworks and platforms are compared among each other based on similarity, data mining task support, parallelism, distribution, streaming data processing support, fault tolerance, security, memory types, storage systems, and others. Similarly, the algorithms are grouped on the basis of parallelism type, scalability, streaming data mining support, and types of data managed. We have also provided taxonomies on the basis of data mining techniques such as clustering, classification, and association rule mining. We also have attempted to discuss and identify the major applications of cloud data mining. The various taxonomies for cloud data mining frameworks, platforms, and algorithms have been identified. This article aims at gaining better insight into the present research realm and directing the future research toward efficient cloud data mining in future cloud systems.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computer systems organization** → **Cloud computing**; • **Theory of computation** → *MapReduce algorithms*; • **Information systems** → **Data mining**; *Data analytics*; • **Computing methodologies** → *Machine learning*; *Machine learning approaches*; *Parallel algorithms*; *Distributed algorithms*;

Additional Key Words and Phrases: Review, survey, taxonomy, framework, data mining, machine learning, distributed computing, cloud data mining (CDM), big data, big data analytics, data science, cloud computing, parallelism, graph mining, volume, velocity, variety, clustering, classification and association rule mining

ACM Reference format:
Hrishav Bakul Barua and Kartick Chandra Mondal. 2019. A Comprehensive Survey on Cloud Data Mining (CDM) Frameworks and Algorithms. *ACM Comput. Surv.* 52, 5, Article 104 (September 2019), 62 pages.
https://doi.org/10.1145/3349265

This material is an outcome of research work carried out in collaboration with Jadavpur University, Dept. of Information Technology, Kolkata.
Authors' addresses: H. B. Barua, TCS Research and Innovation Lab, TCS Ecospace, Action Area 2, New Town Chakpachuria, Kolkata-700156, West Bengal, India; emails: hrishav.smit5@gmail.com, hrishav.barua@tcs.com; K. C. Mondal, Department of Information Technology, Jadavpur University, Sector III, Salt Lake City, Kolkata 700106, West Bengal, India; email: kartickjgec@gmail.com.

0360-0300/2019/09-ART104 $15.00
https://doi.org/10.1145/3349265

1 INTRODUCTION

Looking into the state of the art in *Classical* and *Cloud data mining* [68] techniques, it is to be noted that much work has been accomplished. Data Analysis and Science has gained much importance from the past two decades. Pattern and interesting data extraction is a nontrivial task in the cases of image-processing, Bio-Medical Analysis, Artificial Intelligence, Geo-Spatial Analysis, Astro-Physical Analysis, Genetic Analysis, and many more [140]. Data is grouped on the basis of some similarity and are separated on the basis of dissimilar attributes [140]. Classical Data mining paradigms [123] are basically recognized in the form of *Descriptive* analysis, *Associative* analysis, *Discriminant* paradigm, and *Predictive* analysis. Cluster analysis falls under Descriptive paradigm and Associative Rule Mining comes under Associative paradigm. Classification comes under Discriminant and Predictive paradigm and Regression is considered under Predictive paradigms.

Cloud computing [141, 143, 162] is a newer concept and has its roots in network-based services. It is a platform rather than a paradigm, consisting of computing and storage resources. The Cloud platform, as we call it, is capable of harnessing the power of hardware and software services, such as storage, processing power, software, and network infrastructures [142]. Vendors host resources in cloud in the form of services and clients pay for the services, rent services, and use for their processing needs. With the rise of Big data as a concept [120–122, 124], the demand for processing resources has also grown exponentially. This facility of cloud platform has brought the organizations and clients to advantageous positions as they do not need to buy resources permanently but can rent them on demand. The processing needs for huge digital data can be easily fulfilled this way. Other than this, the Classical data mining techniques are enhanced and parallelized as per data and tasks over distributed cloud computing environment. Figure 1 is an overview of cloud computing services in hierarchy. It shows the relationships between different services and their link with each other. Figure 2 shows the growing popularity of big data mining and cloud computing as compared to classical data mining over the past 10 years (as per Google Trends). Google provides 346 million records on Big data and 140 million on Cloud computing as compared to 83 million on Data mining. Cloud data mining is a new term, often used to refer to data mining using cloud computing services. Google returns around 50 million records on Cloud data mining and 76 million on Big data mining. The prevalent cloud services are *SaaS-Software as a Service*, *PaaS-Platform as a Service*, *IaaS-Infrastructure as a Service*, and *Cloud Storage* Services. *DSaaS-Data Security as a Service* [87] and *RaaS-Resource as a Service* [166] can also be considered as services.

Motivation: With the advent of high-performance computing, distributed systems, and cloud data processing, big data mining has been easier than before. It has become evident that the future is much awaited to witness the real power of Big Data and its processing over the cloud platform as already stated by Talia Domenico et al. in References [68, 73] on distributed data mining over the grids in cloud. While going through the recent advancements in cloud computing and data mining, it has been observed that the various paradigms and frameworks of data mining in cloud have been widely acknowledged by researchers and are utilized in varied forms of analysis of digital data. This review article has been presented, taking the goal in mind that a timely survey regarding the cloud data analytic frameworks and classical data mining enhancements (for this purpose) can serve as a guiding literature. Researchers, data analysts, data scientists, and computer experts can further direct the researches in fruitful paths. This literature review has been supported by many tables and graphs for better understanding of the current state [overview in next section]. Figure 3 shows a typical setup of data mining in cloud environment, which depicts the use of classical algorithms atop cloud frameworks in cloud setup. Also, we have put forward references (in Table 1) to some of the existing related survey papers published in journals such as Information Systems by Elsevier [161], Journal of Big Data by Springer [172, 177], IEEE Communications Surveys & Tutorials [191],

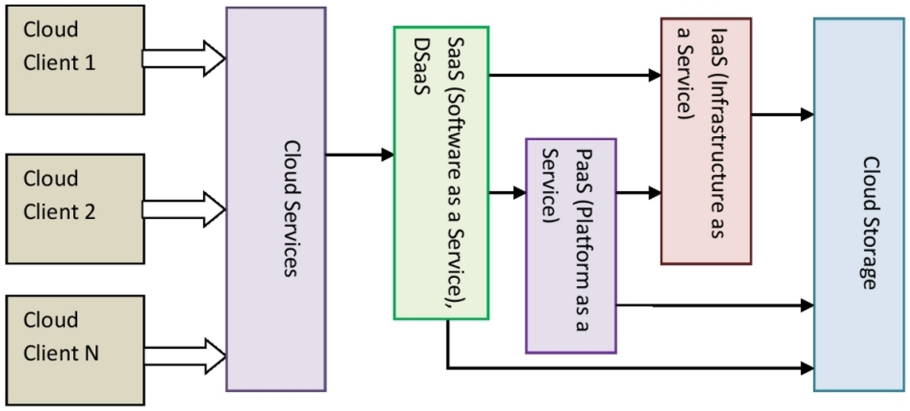

Fig. 1. Overview of cloud computing services.

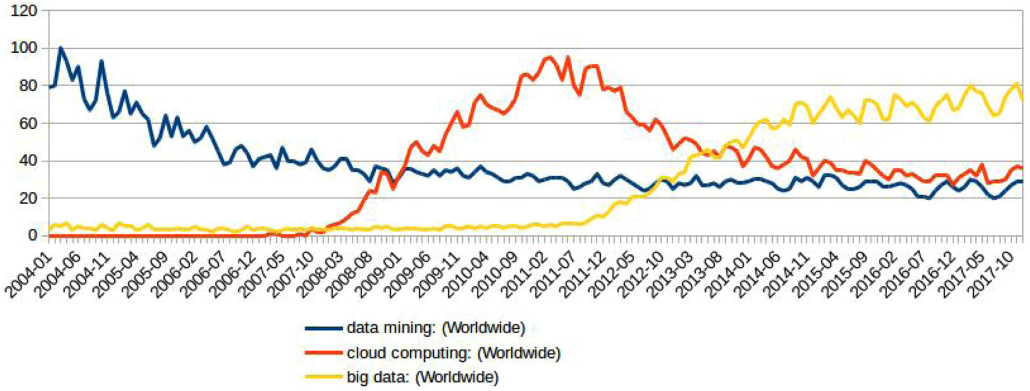

Fig. 2. Popularity of big data and cloud computing compared to classical data mining.

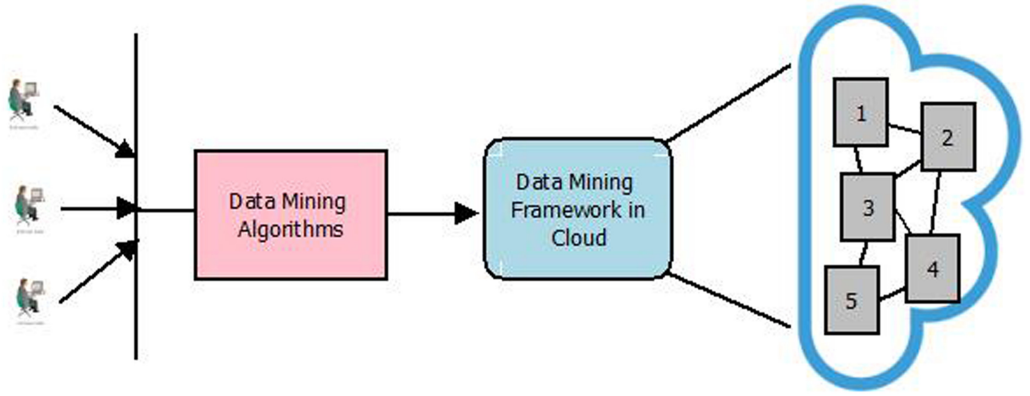

Fig. 3. General framework for cloud data mining.

Table 1. Comparison of Existing Surveys

Existing surveys	Summery
Shakil et al. (2016) [168]	+Summarizes the cloud processing platforms and cloud storages. It also reviews big data processing in cloud. The security-related issues are also discussed. -This survey provides more focus on cloud infrastructure itself rather than big data analytics in cloud.
Nekvapil V. (2015) [169]	+Provides an overview in the importance of using cloud environment in data mining. It also discusses the various cloud-based systems available. The functionalities of such systems for big data analysis have been reviewed. -Proper categorization of Big data mining algorithms and Cloud Platforms is missing.
Marjani, M. et al. (2017) [170]	+Gives an introduction to big data analytics for IOT applications using cloud platform. -Lacks comprehensive discussions on all aspects of Big data analytics in cloud.
Hu T. et al. (2012) [171]	+Puts forward the issues and challenges of parallelism and distribution of data in cloud. The different data mining tasks such as clustering and classification have been surveyed briefly. -All the important algorithms finds less references.
Chun-Wei Tsai et al. (2015) [172]	+Gives an extensive review of cloud big data mining frameworks and platforms. -It lacks discussion about algorithms and techniques for cloud data mining.
Ali Seyed Shirkhorshidi et al. (2014) [173], Btissam Zerhari et al. (2015) [174], Amin Mohebi et al. (2015) [175], Adil Farhad et al. (2014) [176], Hanghang Tong et al. (2013) [178]	+Puts forward survey of parallel big data clustering in cloud frameworks. -Classification and Association Rule mining algorithms for cloud are not covered.
Dilpreet Singh et al. (2014) [177]	+Gives a survey on big data mining and analysis using cloud platforms on the basis of scalability, fault tolerance, iterative processing and some more parameters. -Algorithms and application are not covered.
Ibrahim Abaker Targio Hashem et al. (2015) [161]	+Presents the relationship between big data and cloud computing. It also briefly summarizes some of the cloud data mining platforms. Application areas of cloud data mining have been placed. -Less information and discussions about the algorithms for data mining in cloud
Mohammadi, Mehdi, et al. (2018) [191]	+Presents a comprehensive survey on Streaming high velocity fast Data Analytics for Big Data in IOT devices and Deep learning applications in Cloud. -But proper classification of Frameworks and Algorithms are not shown and focus is on Streaming data only.
Prasad, B.R. et al. (2016) [193]	+Discusses about the Platforms and Services of Stream Data mining. States about the performance evaluators for Clustering and Classification algorithms in high velocity stream mining. -It only focuses on Velocity of Big data concept in Cloud environment.

+signifies advantage and -means disadvantage of the said survey in table.

and more that will help the readers to investigate different perspective of this topic. A comparison of the existing related surveys has also been given in the Table 1.

Uniqueness of This Survey: Our survey comprehensively discusses the major cloud processing frameworks and platforms with the perspective of data mining tasks and techniques such as Clustering, Classification, and Associative Rule mining. We have surveyed the major algorithms implemented in each class and grouped them on the basis of parallelism type, scalability, streaming data mining support, and types of data managed. Application area of different data mining tasks in cloud have been discussed along with optimization techniques for various cloud data

mining frameworks. Security-related researches on cloud data mining has also been placed. The frameworks are also grouped on the basis of parallelism (i.e., task, data and graph parallel system), memory and storage systems, distribution, streaming/batch processing, and others. Relationships between different surveyed frameworks, data mining tasks, and respective algorithms have been placed. Also, we finally show which framework is suited for which data mining tasks, algorithms, and applications.

Organization of the Survey: This review article is managed into the following sections. Section 2 presents all the frameworks for cloud data mining in subsections. The frameworks have been placed chronologically and compared theoretically. The current advancements in performance optimization of the frameworks have been discussed with references. The major platforms and services for Cloud data mining and machine learning have also been summarized with velocity aspects of Big data. Section 3 summarizes the algorithms and paradigms reviewed theoretically under three different subsections on the basis of mining models, i.e., Clustering, Classification, and Associative Rule Mining. The various algorithms have been categorized further into subsections and placed under them. Another section accompanying the three main subsections is placed to show researches in cloud data mining algorithms. Section 4 briefs about the security-related research in cloud data mining scenario. Section 5 presents future prospects about CDM. Some references regarding parallelism, distribution, quantum, and approximate computing in cloud have been given. Section 6 is a briefing about the cloud data mining application areas. Finally, the review has been concluded in Section 7 with a note on the current scenario and future research directions.

2 OVERVIEW OF FRAMEWORKS RECOGNIZED FOR CLOUD DATA MINING

CDM or *Cloud Data mining* frameworks have gained popularity in the current decade for mining data in cloud. Framework, as the term signifies, is a way of execution of process. Data mining in cloud environment needs to be accelerated by some specific setup or way to be executed efficiently. Such setups are commonly termed as frameworks. Frameworks are generally designed with respect to the demand of the processes that should run on it. With the perspective of data mining in cloud, these frameworks need to be designed keeping certain parameters in mind. Parallelism, distribution, and memory management [88, 164] are the primary focus areas of such frameworks. Cloud environment can be highly heterogeneous and diversified, so it is of utmost importance to design an optimized setup. Parallelism can be achieved in many ways: data parallelism, task parallelism, and graph parallelism. Memory can be shared or local. Security of organizational data while performing data mining tasks is also a concern. As far as the frameworks are concerned, many underlying technical ways have been put forward by various researchers to realize these frameworks in a practical realm. Underlying storage systems and memory implementations are the major attributes that distinguish these frameworks from each other. In this regard, the major frameworks identified in this review are: Map Reduce [1, 2], Distributed Graph Lab [11, 12], Sector Sphere [6], and Parameter Sweeping [10]. Ensemble [13]: Data is partitioned into various subsets. The subsets are processed in parallelism. Ensemble of classifiers are created basically using divide and conquer policy. The algorithm is run on each sub-part and the results are merged to form the global classifiers. This can be integrated with Map-reduce or Sector Sphere framework in cloud. Sampling is the technique of taking a sub set of the entire data and building the mining model on top of it. It is easier and efficient than building it upon the entire data. The parameter that needs to be considered is the optimum sample size for the model to be formed accurately.

Figure 4 has placed a categorization of CDM frameworks. Map-reduce framework [1, 2] has been extended to Hadoop [3], Spark [9], FSBD (Framework for Scheduling Big Data mining) [94], and Airavat [59]. Airavat [59] is a Map-reduce-based framework based on the security aspect of data and privacy of users. Pegasus [95] is a graph mining-based framework developed atop Hadoop

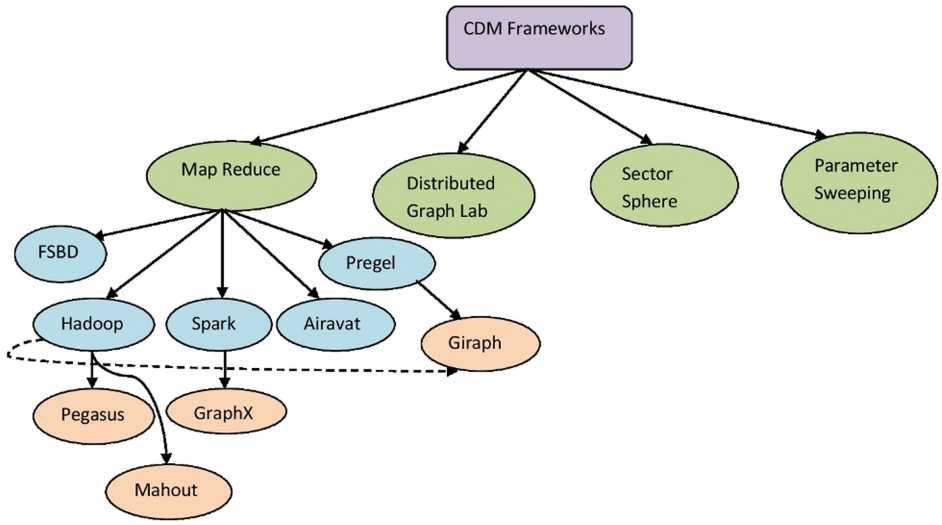

Fig. 4. Taxonomy of different cloud data mining frameworks.

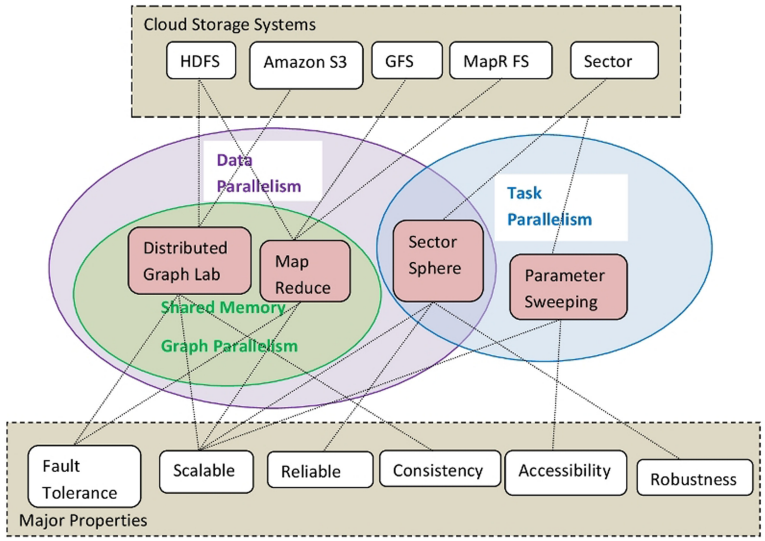

Fig. 5. Relations between cloud data mining frameworks, their properties, parallelism, and storage systems used.

[3]. Pregel [96] and Apache Giraph [97–99] are graph mining frameworks inspired by Map-reduce mechanism. GraphX [100–102] is a graph parallel framework built upon Spark. Apache Mahout [131, 132] is a framework inspired primarily by Hadoop Map reduce [1–3]. Figure 5 is the grouping of major frameworks on the basis of parallelism they provide along with memory management schemes. The storage systems and properties for each framework have been clearly presented. As already mentioned parallelism can be data centric, task centric or graph centric. Parameter Sweeping [10] is a purely task parallel system. Sector Sphere [6] exhibits data as well as task parallelism. Unlike Parameter Sweeping [10], Map Reduce [1, 2] exhibits data parallelism and graph parallelism. Distributed Graph Lab [11, 12] is a graph parallel system. Graph parallelism is a special

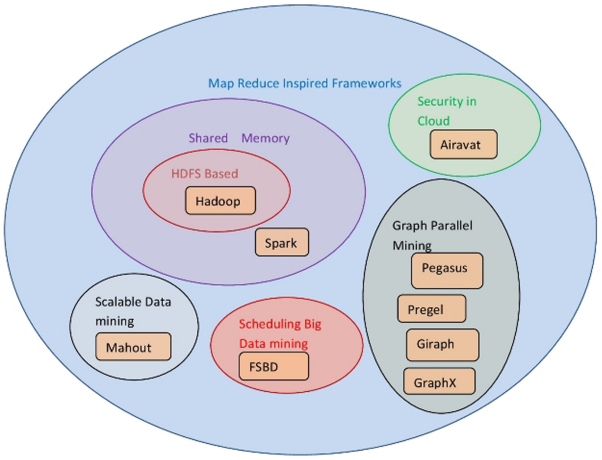

Fig. 6. Grouping of map-reduce-based cloud data mining frameworks.

case of data parallelism. Apart from this Distributed Graph Lab [11, 12] and Map Reduce [1, 2] practices shared memory implementations. Figure 5 also clearly shows the cloud storage systems used by the frameworks and properties they exhibit. Figure 6 is depicting relation between the renowned extensions of Map Reduce [1, 2]. Apache Hadoop [3], Apache Spark [9], Pegasus [95] and Airavat [59] are the major extensions. FSBD [94], Pregel [96], Giraph [97–99], GraphX [100–102], and Mahout [130, 131] are the minor extensions. This will be discussed later in this section.

2.1 Map Reduce

Map-reduce [1] programming architecture was a breakthrough in Big Data processing field, introduced by Google in 2004 [2]. This architecture is inspired by normal database functions that we use in day to day data retrieval activities. The paradigm is composed of a Mapper function *Map()* and a Reduction function *Reduce()*. Map function can be resembled with the sorting/searching functions performed on records in database table and Reduce can be thought of as the aggregate and filtering function that we perform on records. The difference is in the volume of data Map Reduce works.

This framework supports parallelism and high degree of data distribution. Motivated by Large-scale data processing and utilization of maximum resources such as thousands of CPUs and fault tolerance, the approach is widely used over the cloud. Briefly describing the working principle of this technique, the Mapper and Reduction functions are stated below.

Mapper Function is responsible for taking the entire Big Output and feeding it into the master node, which in turn divides it into many sub-parts to be solved individually by nodes (can be n number of computing nodes). The solved problem is sent back to the master node. The Reduction function aggregates the locally solved problem output in one way or other and finds out the global answer to the original Big Dataset. To perform these actions a *Key-value* pair is used by the system. *Key-value* pairs are generated from input data. Each computing node is fed with some *Key-value* pairs, which are processed to further generate new *Key-value* pairs. The *Key-value* generated is mapped to the node that should work on it hence providing the generated data associated with that key value. The *Key-value* in each node is being processed by reduce function exactly once. Then the output is collected by the framework and sorts it by *Key-value*. The storage of this framework is taken care by distributed file systems. Figure 7 is a depiction of Map-reduce [1, 2] framework consisting of Mapper and Reduction functions as described above.

Coming to the performance of this framework, it is reported to have attained high degree of data parallelism unlike Parameter Sweeping framework where high degree of task parallelism is

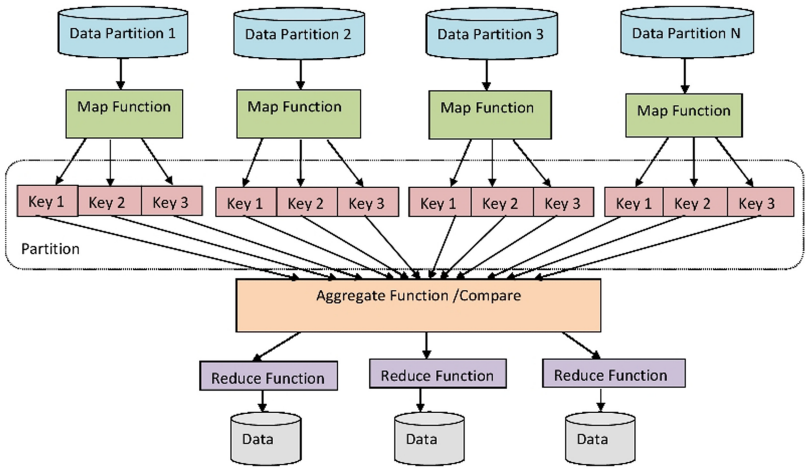

Fig. 7. Overview of map-reduce framework.

obtained. This framework is used in many classification and clustering tasks such as Apriori implementation and spatial data mining implementations. Map-reduce technique can efficiently be used to find patterns in large files in UNIX environment applying Grep. Now a days Geospatial analysis is coming to spot light with the growing popularity of raw data from satellites. Google implements Geospatial analysis of its Maps with the Map-reduce framework. Webpage access frequency is also being counted with Map Reduce. This framework is capable of processing document or set of documents and mining out important patterns or words out of it. A major drawback of this framework is its inability to process such data where computing a value depends upon the computation of other value. Heterogeneity is another problem faced by large-scale Map-reduce implementations. If the features and parameters of the large number of participating nodes is different from the other, then there can be serious performance degradation of overall efficiency and latency of the mining applications. To improve this situation LATE (Longest Approximation Time to End) scheduler was utilized in Map-reduce model [4]. LATE is reported to improve Hadoop response time 2 fold in clusters of hundreds of nodes and virtual machines in EC2 (Amazon's Elastic Compute Cloud) [5].

2.1.1 Major Frameworks Based on Map Reduce. Map-reduce framework has been considered as a basis for many data analytics tasks over cloud. Due to its scalability and parallelism, it has attracted many practitioners to implement many data mining algorithms using its functions. Also, many researchers and innovators have proposed many extensions of Map Reduce for various purposes and improvements over the original one. With this regard, the major extensions of Map Reduce can be considered as: Apache Hadoop [3], Spark [9], Pegasus [95], and Airavat [59]. Table 2 provided below compares all the major Cloud data mining frameworks using Map-reduce technology.

Hadoop: Apache Hadoop [3] framework has its backbone in Map-reduce architecture. It is open-source in nature and is capable of supporting development of applications that can run in parallelism. Hadoop implementation has been featured with accessibility, robustness, simplicity and scalability. It uses cloud services such as hardware, processing power and storage for processing. Fault Tolerance is also taken care by hardware itself and new nodes can be added as per requirement to increase parallelism and decrease execution time. *HDFS* or *Hadoop Distributed File System* is the underlying storage of Hadoop Map-reduce system similarly as Google employs *GFS* or *Google*

Table 2. Comparison between Major Map-reduce-based Cloud Data Mining Frameworks

Frameworks with date of initiation	Major Algorithms implemented	Optimized for	Advantages	Disadvantages	Features and Underlying Storage	Parallelism Type	Principle Used
Spark (2009–10) [9]	Frequent Item set Mining using Apriori, DB-SCAN	Any kind of data flow model, huge data, iterative and interactive processing	In Memory Processing, Fault tolerance, Acceptable Latency	Doesn't have its file management system so has to be integrated with HDFS, GFS or others	HDFS/MapR FS	Data Parallelism	RDD-Resilient Distributed Datasets
Hadoop (2009–10) [3]	Apriori, Classification and Clustering, Spatial Data Mining	Any kind of mining application running in high degree of data parallelism	New nodes can be added as per requirement to increase parallelism, The data is actually stored in each node and HDFS API shows the single file system view for end users	Inability to process such data where there are dependencies between computing values. Heterogeneity is another problem w.r.t large-scale implementations. There can be serious performance degradation of overall efficiency and latency of the mining applications due to heterogeneous nodes	HDFS, accessibility, robustness, simplicity and scalability	Data Parallelism	The computation is scheduled near the data using HDFS features. Hadoop uses HBase for storage
Pegasus (2009) [95]	Web mining Algorithms	Graph mining [163] and mining billion scale graphs	Large graph mining packages, Parallel algorithms atop Hadoop, Open source	Not found any	Scale-up and good runtime	Graph Parallelism	The framework is based on matrix vector multiplication
Airavat (2010) [59]	k-Means, Naive Bayes	Security of Data while mining in cloud	Privacy-preservation of data and users, accuracy of data mining tasks	Overhead introduced in general Map-reduce architecture, Limitations in security of malicious and untrusted computation providers	Good Parallelism ensuring security, Distributed File system of MapR	Data Parallelism	Based on Mapper and Reducer functions as in Map Reduce

File System as its underlying storages [71]. It is efficient and reliable using number of hardware in clusters. The data is actually stored in each node and HDFS API shows the single storage view for end users. The file replication and data replication procedures are carried out to maintain reliability and consistency. Hadoop has the facility to optimize the data and value transfer between nodes using data locality. The computation is scheduled near the data using HDFS features. Big tables that are used by Map-reduce framework are replaced by Hadoop with HBase [112–115]. ApproxHadoop [165] is an add on to Apache Hadoop Map-reduce framework, which uses approximate computing techniques to reduce program latency by 32× along with energy consumptions with a small percentage of tolerable error. Data sampling, task skipping, and multiple approximate versions of code are the techniques used in this framework for approximate computing.

Spark: Spark [9] is an advanced version of data mining framework that has been proposed by Matei Zaharia et al. in University of California, Berkeley. Most of the frameworks, such as Map Reduce and its variants like Sector sphere and Hadoop engine systems, deal with acyclic data flow models, which does not satisfy the needs of many data mining application and techniques. Spark is designed in such a way that it can overcome the shortcomings of the general Map-reduce-based frameworks and can apply the advantages of this frameworks. Scalability and Fault Tolerance of Map reduce is replicated in this new framework. The framework works upon the concept of *RDD-resilient Distributed Datasets*.

Coming to the design and development of this framework, it is to be noted that it is developed with Scala Java VM. The concept of *RDD* is utilized to generate parallel operation like Map Reduce by caching in memory in nodes. *RDD* can achieve fault tolerance by implementing replication

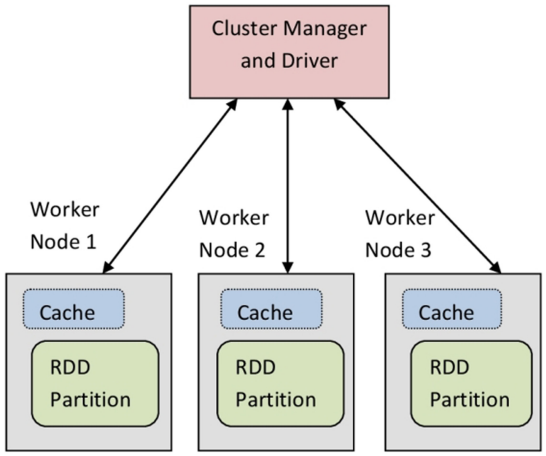

Fig. 8. Overview of spark framework.

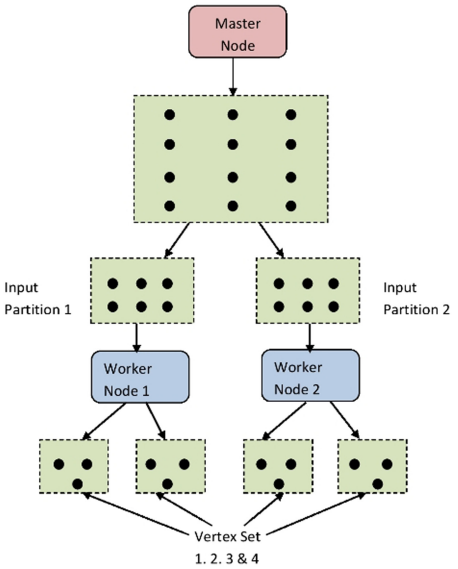

Fig. 9. Overview of graph mining in Pegasus.

technique from other *RDDs*. Users have the advantage of defining *RDDs* and other functions for parallel operations in cluster. Two shared variables used in Spark are: broadcast and accumulators. Figure 8 describes the working of Spark framework.

Spark has been reported to outperform Hadoop for some data mining in machine-learning tasks and is also reported to have a very promising latency and throughput. Spark Streaming [189] is an added advantage over core Spark model. It has the capability of processing streaming data rather than batch processes. The various applications and algorithms of Data mining and Machine Learning can be run atop this framework written in Java and Python. StreamDM [190] is an advanced library (developed in Huawei Noah's Ark Lab) for Data mining tasks implemented atop Spark Streaming. Spark Streaming lacks open source Data mining libraries, which is taken care by StreamDM. It is easy to use and highly extensible. IncApprox [188] is a framework built atop Spark [9] for Green Data Analytics using Incremental and Approximate computing paradigms. Data sampling and Memoization techniques are being used. It has been successfully used in Twitter data stream analytics and network traffic monitoring applications.

Pegasus: With the advancements of digital data, the process of mining such data has grown immensely. With the arrival of social networks and data in such area, the linked data, it is very important to have robust mining techniques for linked data as well as relationship data in net.

Pegasus [95] is a framework of cloud data mining implemented over Hadoop Map Reduce [3] for mining large-scale graphs in the cloud big data realm. The framework is based on *matrix vector multiplication*. Very good scale up and runtime have been achieved with this framework for graph mining. Spectral clustering tasks can be efficiently performed using Pegasus framework in cloud. Figure 9 (black dots are vertices in graphs) and Figure 10 are the pictorial representation for Pegasus framework as described below.

According to Generalized *Iterative Matrix-Vector multiplication* (GIM-V), multiplication of matrix M and vector v can be broken into three distinct operations: Multiplication of M and v, summation of results and re-establish the value of vector v with the newly computed value. These three operations can be used for huge scale graph mining.

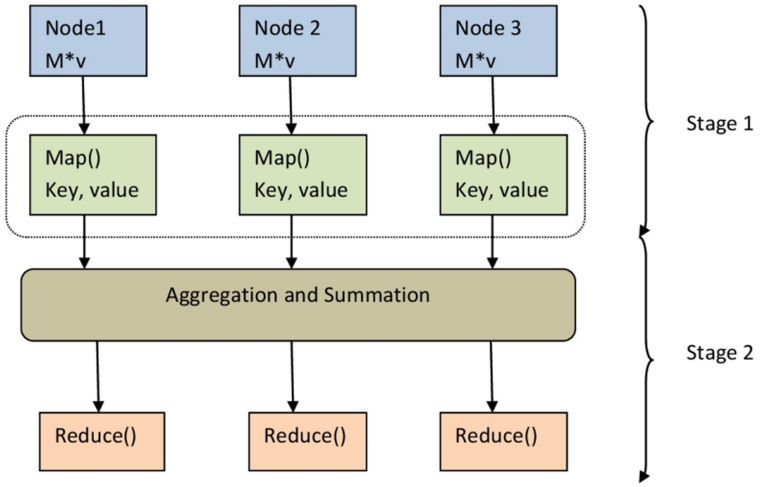

Fig. 10. Overview of graph mining in Pegasus using Hadoop.

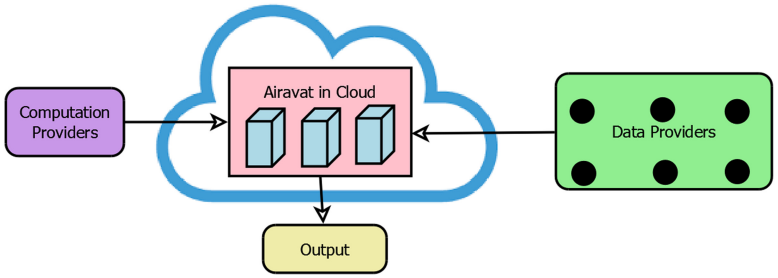

Fig. 11. Overview of Airavat framework.

In Hadoop environment, Pegasus works in a distributed mode for the above mentioned operations. The operation is performed in two stages: The first stage combines the columns of M with rows of v. *Key-value* pairs are generated with key as the source node and value as the combined results from multiplication. The output of this stage is fed as the input of the second stage where the results are combined fully by summation. Assignment of the vector value to newly computed value is also performed.

Airavat: Airavat [59] is a novel model for cloud data mining well equipped with security and privacy facilities for cloud data of users and organizations while mining tasks are carried out. Amazon's Big data has been targeted to implement the framework.

The framework is three layered architecture: Data Provider, Computation Provider and Security Provider. Figure 11 describes the overall high level model of Airavat framework. The data providers set some security-related restrictions as per organizational data policy. The computation providers are those groups of people who actually provide the algorithms and paradigms to process the data as per requirement. The data providers in cloud can have labels of security layers. The computation on such data can be done and the labels can be made a union to the result. A differential mechanism [60–63] has been adopted to ensure such security in Airavat. The Airavat model can set the bounds in the computation providers to ensure the proper sensitivity of the functions. The *Map-reduce distributed file system* storages are used as an underlying storage.

Table 3. Comparison between Minor Map-reduce-based Cloud Data Mining Frameworks

Frame-works with date of initiation	Major Algorithms implemented	Opti-mized for	Advantages	Disadvantages	Features and Underlying Storage	Parallel-ism Type	Principle Used
FSBD (2014) [94]	K-means Clustering	Huge data sets and Big data clustering	Can be used as SAAS over the cloud, Minimizing execution time, Scalable	Not Known	Distributed Architecture, DFS	Big Data Parallelism	Have Master Worker, Compute Worker and Resource manager for scheduling and managing clustering tasks
Pregel (2010) [96]	Clustering Algorithms	Graph Mining [163] applications such as social network analysis	Efficiency, Fault tolerance and Scalability	Not applicable if terabytes of RAM/Main memory is required for processing as computation state is kept in RAM itself, Master node not involved in computation	Local Storage and RAM of worker nodes	Graph Parallelism	Master node partitions graphs as set of vertices and edges and input to worker node for computation
Giraph (2015–16) [97–99]	K-means Clustering	Graph mining [163] and mining billion scale graphs	Large graph mining packages, Parallel algorithms atop Hadoop, Open source, Master node computation happens	Not found any	Scale-up and good runtime, sharded aggregators, edge-oriented input, out-of-core computation, composable computation, storage- HDFS and Hive tables	Graph Parallelism	Iterative graph processing via compute() function by user: Supersteps
GraphX (2013–14) [100–102]	Distribute Genetic Algorithm for data mining	Large Graph Computation	Low cost computation and good fault tolerance and features of Graph and Data parallelism	Not fit for relational databases (researches are going on)	Expressiveness, performance, and ease of use	Graph Parallelism	Programming abstraction called Resident Distributed Graphs (RDGs) is used and it significantly simplifies graph loading, construction, transformation, and computations
Mahout (2017) [130, 131]	k-Means Clustering, Spectral Clustering, Fuzzy k-Means	Scalable Algorithms	Scalability, High Performance	Out of core operational support missing	Versatile and robust	Data Parallelism	Linear algebra and statistics are the basis

Although Airavat is a security expert framework, it has some limitations. The malicious computation providers delivering untrusted Mapper functions cannot be treated by the setup. It is an ideal model to ensure security policies for data providers. The SELinux [126] kernel has been used for implementing Airavat's access control and security.

2.1.2 Minor Frameworks Based on Map Reduce. Map-reduce-based frameworks are increasingly evolving through the recent years and it is a must to mention all the frameworks with minor extension and features so that the researchers can have a complete review of the scenario. The minor extensions under Map-reduce framework are: FSBD [94], Pregel [96], Giraph [97–99], GraphX [100–102], and Mahout [130, 131]. Table 3 given above is a table of comparison for minor Cloud data mining frameworks inspired by Map reduce.

FSBD: FSBD [94] or *Framework for Scheduling Big Data mining* in cloud is a setup that can be considered as a SaaS. *Software as a service* is a facility provided by cloud distributors. FSBD

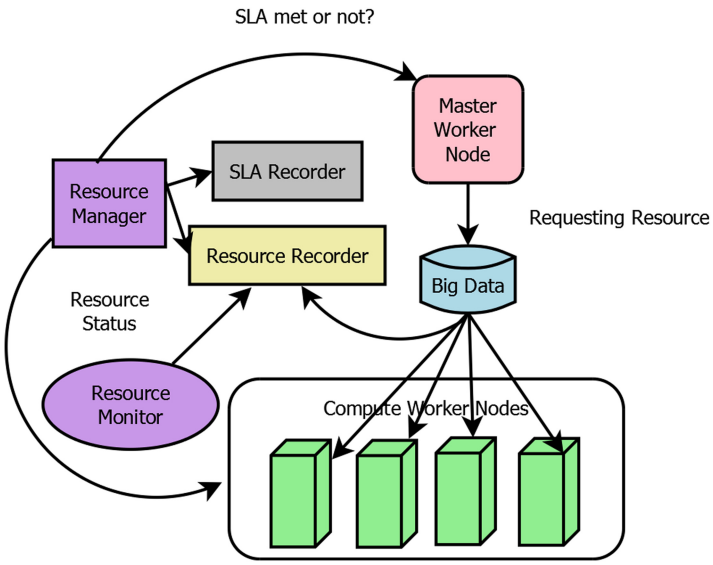

Fig. 12. Overview of FSBD architecture.

can be considered as a software system that is capable of implementing data mining tasks as per the parameters and needs of any researcher or data analyst. Data mining task distribution and scheduling is properly ensured by the framework that we call a software service. Cluster-based mining of data has been previously carried out in cloud environment [127, 128].

The basic concept of FSBD [94] is the real time application of cluster data mining with ample monitoring of the same. The components of FSBD can be classified as: Master Worker, Computing Workers, and Resource Managers. Figure 12 is the pictorial representation of typical FSBD architecture. The first component has the responsibility to schedule the big data mining and its distribution is also taken care in parallel. The Computing workers are the actual data mining components that do the data processing locally after the distribution and allocation is completed. The Resource manager is assigned the task of managing resources and allocating resources along with monitoring. The resource manager has a special unit called resource recorder that checks the status of resources. A SLA recorder is also an important entity as it checks if the Service Level Agreement for resources are matching or not. The master node allocates data partitions to computation nodes using scheduling algorithms and attains load parallelism [129].

K-means clustering has been successfully implemented in this framework. The framework has been experimented and reported to have diminished execution time of clustering algorithm without loss of data cluster quality over the net. Also with the proper use of quality metrics dynamic redistribution of data can be incorporated [130].

Pregel: The Pregel [96] is a graph processing and mining model developed in 2010 by Grzegorz Malewicz et al. in Google. Now a days the social networks and web site relationships (Facebook, Twitter, Research Gate, etc.) have been the center point of attraction for the data analysts and scientists. The mining of graphs (edges and vertices) in a social network database is a nontrivial task in data mining context. The iterative programming model is used to process graph vertices in Pregel framework. The messages can transcend iterations and alter graph topology as per need.

Coming to the architecture of Pregel model, the Pregel partitions the entire graph structure of the data into sub parts consisting of the vertices and edges. Vertex IDs are used to track vertices

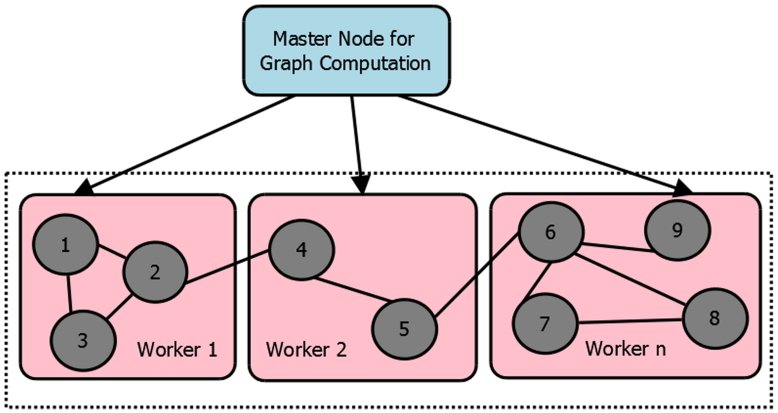

Fig. 13. Overview of Pregel architecture.

assigned to partitions and machines that actually own them. The execution of the Pregel framework starts with the execution of the worker nodes in the cloud. The master worker node does not necessarily process any partition of the graph but is a coordinator of the entire distributed activity in cloud. The master worker assigns partitions of the graph to the worker as per the number of partitions to be created. The master worker node invokes *compute()* function for all the active vertices of partition. Figure 13 is an overview of Pregel working with partitioned graphs.

Giraph: Giraph is a graph processing model inspired by Pregel [96] and built atop Hadoop [3] but is open source. The architecture is similar to Pregel [96]. The model is inspired by the vertex-graph system. The vertexes can store data of entities and edges are capable of storing any kind of relationship between the entities. Facebook uses this model for processing connection between its users and their network. Graphs with huge amount of edges can be processed using higher scalability and flexibility. Giraph has some minor extensions on Pregel framework [99]. The computation of graphs takes place in iterations [96, 98, 99]. These iterations are called super steps. A *compute()* method, as defined by user, is called by the active vertex in the superstep. The *compute()* function is capable of receiving messages of previous superstep and send messages to other vertices along with computation of the current vertex and edge values.

GraphX: GraphX [100–102] is a graph parallel framework inspired by Spark in general [9]. The framework is an embedded graph parallel framework. The graph data flow operations in GraphX can be handled by some data flow operations such as join, map and group-by. The graph operations such as joins are done in distributed manner. So, the GraphX is capable of low cost fault tolerance. The graph is partitioned into dependent data in graphs such as relationships with neighbors. The partitions are distributed over various computing resources in cloud. The dependencies of each partition with other are taken care by computation over the edges of the graphs and communication between the computations on the partitions in the compute nodes. A distributed Genetic Algorithm [103] has been implemented using GraphX framework in cloud. This framework has been reported to perform better than graph-lab [11, 12] and Spark [9].

Mahout: Apache Mahout [131, 132] is a framework inspired primarily by Hadoop Map reduce [1–3]. Mahout is a computation friendly model providing java libraries. This framework is capable of driving clustering and classification tasks of data mining. The library that forms the basis of this framework is linear algebra and statistics. It is also flexible enough to be implemented in a single node non Hadoop architecture.

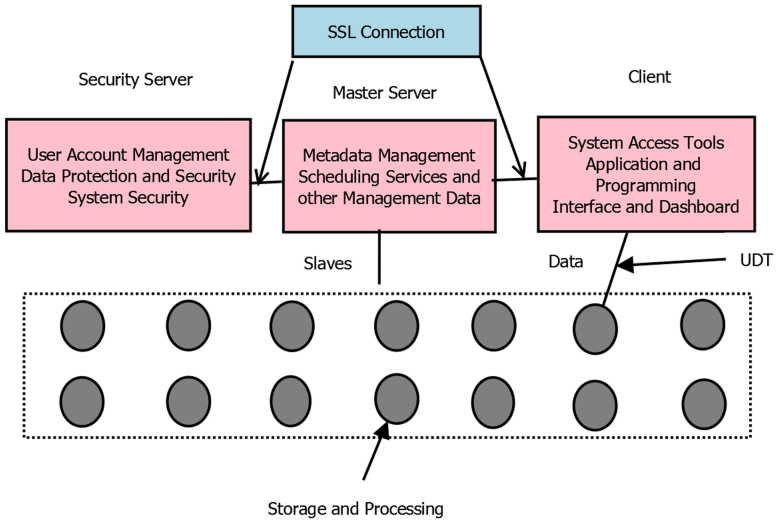

Fig. 14. Overview of sector sphere framework.

2.2 Sector Sphere

In the continuous process of evolving the distributed data processing systems and software, computer researchers have come up with some really novel approaches. Sector-Sphere is one such framework identified by researchers for distributed storage and processing of data over the cloud. The *DaaS* and *IaaS* of cloud are being used to implement this architecture. The name Sector-Sphere has its meaning in two underlying concepts. *Sector* refers to the distributed file system of user space in each node for storing files that can be uploaded by users. It can store data over the cloud in a large number of computer nodes. *Sphere* is the distributed execution control of the framework. It is capable of processing in-storage massively parallel data stored in sector.

The working of this framework is something like the previously discussed Map-reduce Framework. It marks its difference from Map-reduce framework in its adaptability to user-defined generic functionalities in contrary to Mapper and Reduction functions in Map-reduce Framework. The *Sector-Sphere* framework is allowed to operate on wide area network with high-performance factor.

Coming to the architecture of this framework, it has 4 major components, the Security server, the Master Node, the Client Node, and Slaves. Figure 14 represents the architecture of Sector Sphere framework. Security server takes care of the security policies and user authentication for accounts as well as access controls. The Master node/nodes are responsible for the overall administration of the entire setup and system. It is also responsible for accepting user requests in distributed environment and responding the same. The slave nodes are responsible for processing the stored data files as and when requests are raised. Sector-Sphere computing in cloud is a bit complicated scenario as a whole. High numbers of concurrent connection are generated. The periodicity and frequency of connection is highly random. The data communication is done using *UDT* (UDP-related Data Transfer) [6, 7, 80].

Coming to the performance of this framework, scalability and reliability are the parameters to be considered. The underlying network infrastructure highly influences the performance of this framework. The speed of data communication is directly proportional to the performance of the network and its routing algorithms as well as network set of protocols. High-performance WAN is widely used for this framework [7, 69, 80].

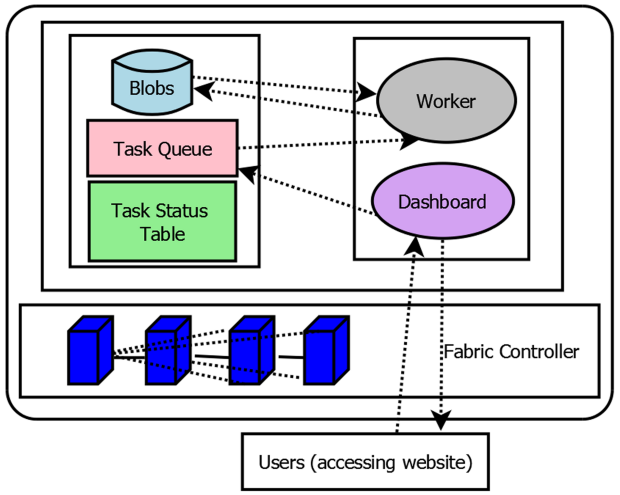

Fig. 15. Overview of parameter sweeping framework.

Association rule mining has been implemented in this framework. This framework has been reported to have performed better than Hadoop Map-reduce framework. Yunhong Gu, Li Lu et al. [8] have successfully utilized Sector-Sphere framework to solve graph BFS problem in cloud data mining. Also, it has proved the superiority of this framework over Map-reduce framework.

2.3 Parameter Sweeping

Parameter sweeping [10] is a concept where data mining algorithms are executed with different set of parameters each time also considering different range of values for those parameters till satisfactory results are gained. The effect of different combinations of parameters and parametric values on the results of data analysis is a key use of parameter swapping. With the number of varied swept parameters and their ranges it can lead to numerous tasks to be executed. But it is very inefficient to serially execute such jobs. So, to overcome this issue the cloud services are utilized. The *SaaS* and *DaaS/PaaS* can be taken into account. The SaaS ensures that any one algorithm can be selected from a list of algorithms for a particular task. The *DaaS* is the storage and processing power that needs to be used as service in the cloud. To attain task parallelism resources must be used over the cloud .This approach has been reported to have a very effective *analysis parallel execution* based on parameters.

This paradigm is practically realized in cloud using Windows Azure. Windows uses *PaaS* to implement this technique. A set of fully fledged servers are applied by Windows Cloud Data Center. Azure is capable of harnessing the power of cloud and it achieves high availability and high scalability, i.e., on demand resource addition. Storage of huge datasets and highly parallel batch processes can be computed by Azure. The working principle of this framework is very simple and robust. The framework is constructed with the following concepts: a set of Blobs, a Task Queue, a Task Status Table, a Pool of *k workers*, and a Website. Figure 15 is an overview of Parameter Sweeping framework. Blobs are used to store huge data to be analyzed and mined. Task queue is taking care of data mining tasks to be performed and the status of such tasks is tracked in task status table. *K workers* are the servers capable of executing data mining tasks. The overall dashboard of the activity from submitting of task, monitoring status to viewing results is done through a website. Now, coming to the importance of parameter sweeping paradigm, the data mining task is run as multiple instances over the same data to be analyzed but with different set of parameters

and their ranges in parallel. The Azure affinity feature is used to place the data as close as possible to the servers. The framework is a combination of compute, storage and fabric controller.

Horizontal scalability is an important trait of this framework. The latency and turnaround time of the activity decreases with the addition of servers to the set. Fair enough speedup is achieved for scalability. Linear speedup is achieved for homogeneous tasks such as classification. SaaS of cloud is efficiently made to use in this framework unlike graph lab where Storage and DaaS were adopted efficiently. Applicability of this framework can be widely seen in Clustering and Classification Tasks. *K-means* algorithm was successfully and efficiently implemented in this framework. J48 was used for classification task effectively. For both this algorithms belonging to Clustering and Classification tasks, the speedup was almost linear and the turnaround time is decreasing drastically with the increase of servers.

2.4 Distributed Graph Lab

Graph Lab/Distributed Graph Lab [11, 12] or Shared Memory Graph Lab is a Cloud Data mining model that is regarded as an efficient model for implementation of some of the most pioneering Data Processing systems and algorithms. It uses a shared-memory implementation to support graph parallel computation of data further ensuring data consistency and high degree of parallelism.

The Graph Lab framework is constructed on three concepts signifying the graph of data, the function used for updates and the sync operation. The program state is stored and tracked by Data Graph. $G = (V,E,D)$ is the representation of this Data Graph. D is Data, widely consisting of Program State, Processing Data and Parameters as required. To take for an example, in execution of a huge data set for cluster detection, we have distributed the dataset into various sites in cloud. The data (D) of the graph lab might correspond to the local clustering results from every site and the edges (E) might correspond to the relationship among the different partitioned datasets (V) fed into the sites. This also leads to data parallel implementation of data mining task. The update function is utilized to manipulate data in vertex and updates the data within the scope of the said vertex. The sync operation is finally used to transform the locally updated data into global form. Dynamic data updating activity takes place in parallelism asynchronously. The problem with synchronous snapshot technique in fault tolerance is that it was not efficient enough and was dependent on single system performance. So, asynchronous snapshot technique was introduced using Chandy Lamport Snapshot design [92, 93].

Now coming to the performance and applicability point of view, this approach is very much acclaimed due to its fault tolerance, scalability and maximum resource utilization in cloud environment. The trade-off between high level distribution and consistency is managed successfully with the data graph, update and sync functions. The degree to which maximum data mining algorithms fit into this model is high. Memory consumption is limited as shared memory realm is used for the computation. Efficient utilization of cloud storage, processing power and infrastructure from cloud as service can be seen. Now, comparing between consistency, performance and throughput, the model is averagely situated. To enhance performance throughput must be compensated by maintaining partial consistency and to have accurate throughput, performance might have to be sacrificed by implementing full consistency using Graph Coloring and Distributed Locking approaches.

This framework has been successfully applied in some common data mining and Machine-learning problems as page Rank, Matrix Factorization, Social Network Analysis, and SVM. As already cited above, this can very well be a good paradigm for spatial data mining application over the cloud. Some algorithms adopting this framework over the cloud are K-Means, DBSCAN and Spectral Clustering. Many Machine-learning and Image-processing tasks have recognized the

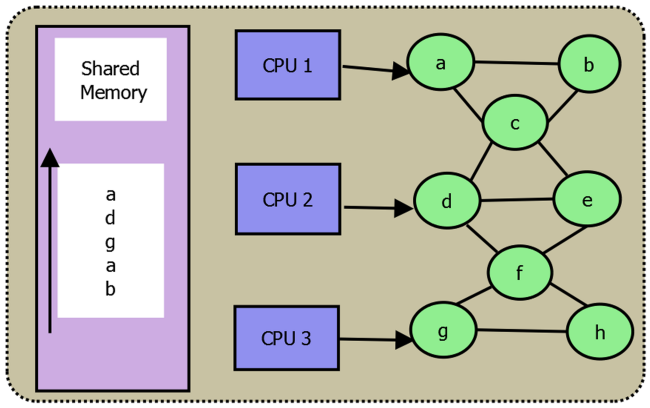

Fig. 16. Overview of distributed graph lab framework.

importance of Graph-Lab model and have successfully applied the same in their capacity. Some of them are Netflix Movie Recommendation, Video Co-Segmentation, Named entity Recognition and EC2 Cost Evaluation. Graph Lab has many add-on libraries for various machine-learning tasks such as Clustering, Computer Vision and Collaborative Filtering. Figure 16 is an overview of Distributed Graph Lab framework. The image is a typical implementation of Graph Lab Framework where *abcdefgh* is the designated graph *G* consisting of the said vertices (*V*). Let the graph lab be distributed over three computing resources or CPUs. Each executing update functions on data of vertices. The left-most arrow defines the direction of scheduler in the shared memory realm. The edges such as *ab* are the relationship between the data of two vertices *a* and *b*. This graphical representation is a basic depiction of any algorithm running in Graph-Lab framework such as any social network analysis algorithm, page-rank or any distributed implementation of *K-Means* over the cloud. Table 4 is the comparison of the major cloud data mining frameworks surveyed in this article.

Tables 5 to 7 compare the discussed frameworks and their properties and also state the application areas and limitations. Table 5 is presented with a summary of all the discussed cloud data mining frameworks. It shows the similarities between the frameworks, data mining algorithms implemented in the frameworks and possible areas where future research can be directed. Table 6 shows the evaluation done on the various frameworks with respect to different properties and features based on this literature survey. Table 7 clearly shows the parallelism supported by each framework.

2.5 Optimizations in Cloud Data Mining Frameworks

Cloud data mining has a promising future, provided the resources are utilized effectively. The survey placed in this literature is comprehensive keeping in mind the current scenario. Along with the implementations of data mining algorithms over the cloud over some frameworks, it is also mandated that the performance of the algorithm is maximum [89]. High-performance computing is not an added advantage anymore but a necessity. In highly scalable cloud setup for mining, inter node communication delay poses a serious threat in performance. The concept of data locality has come to existence to cope with this network latency. Apache Hadoop [3] utilizes this concept to enable the processing of liable data as per proximity of data with the compute node. Zhenhua Guo et al. [72] have put forward some deep insights about data locality concept over Map-reduce framework. Performance Tuning is another aspect to be researched in cloud mining frameworks and paradigms. Setting right values to performance parameters and even considering the right

Table 4. Comparison of Cloud Data Mining Frameworks

Frame-works with date of initiation	Major Algorithms implemented	Optimized for	Advantages	Disadvantages	Features and Underlying Storage	Parallelism Type	Principle Used
Map reduce (2004) [1, 2]	Distributed K-Means, Bi-Clustering Algorithms, MP Apriori, FP Growth, DB-SCAN	Large datasets and acyclic data flow model	Highly Parallelized over a number of compute nodes, Fault tolerance and Scalability	Inability to process such data where computing a value depends upon the computation of other value. Heterogeneity is another problem faced by large-scale Map-reduce Implementations	Shared Memory Buffer used GFS-Google File system	Data Parallelism and Graph Parallelism	Two function used: Mapper Function and Reduction Function used
Parameter Sweeping (2011) [10]	K-means and other Clustering Algorithms	Huge Data Sets and data having variable features, Homogenous mining tasks	Horizontal scalability, Latency decrease with addition of servers	It is not very efficient when workflow of any mining task is considered important unlike Map reduce	User Specific Task Submission, Monitoring and Management dashboard	High Task Parallelism	Separate storage, compute facility lying on top of fabric controller, Multiple instance of a single mining algorithm
Sector Sphere (2008) [6]	Associative Rule Mining Algorithms	In-storage massively parallel data stored in sector storage, user specific customized tasks	Distributed computing, Data can be processed in place, Utilize the bandwidth available on wide area network by UDT, Scalable and Reliable	The performance of the framework is highly dependent on the underlying network infrastructure. The speed of data communication is directly proportional to the performance of the network and its routing algorithms as well as network set of protocols	Execution of user-defined function to whole datasets, Support different routing and networking protocol, Separates storage and compute facility	Data Parallelism and restricted Task Parallelism on demand	Sector refers to the distributed file system of user space. Sphere is the distributed execution control of the framework.
Distributed Graph lab (2012) [11, 12]	K-Means, DBSCAN, Spectral Clustering	Maximum resource utilization, tradeoff between high level distribution and consistency, Serializability	Shared memory Graph Lab notion is extended to support distributed computing, fault tolerance, good Scalability	Framework does not support dynamically evolving graphs and external storage in graph databases. Does not separate storage and compute	shared-memory implementation, ensure data consistency and high degree of parallelism	Graph Parallelism and Data parallelism	The program state is stored and tracked by Data Graph. G=(V,E,D) is the representation of this Data Graph

set of parameters for any mining specific task plays a huge role in optimizing tasks. Novice users generally make wrong choices of values and parameters resulting in unacceptable complexity and performance. Sivnath Babu [74] has investigated the issue and come up with some automation in parameter selection and valuation for heterogeneous tasks. Jahani et al. [75] proposed another technique for performance optimization using B+-Tree for data partitioning and compression techniques for optimizing I/O. Praveen Kumar Lakkimsetti [76] has proposed a sample-optimization technique where a sample of data is processed with different combinations of parameters. The most proficient combination is considered for processing other data volumes. Nezih Yigitbasi et al. [77] proposed machine-learning-based performance parameter valuation and selection for improving Hadoop framework performance. Vasiliki Kalavri et al. [79] placed some performance-related research on Map Reduce and its variants, which the readers can further explore for better insight about the scenario. Much works have been done on Map-reduce and Hadoop platforms but the other major frameworks such as Sector-Sphere, Distributed Graph-Lab and Parameter-Sweeping needs to be attended to improve data processing and parallelism.

Table 5. Summary and Future Research Directions of Cloud Data Mining Frameworks

Cloud Data Mining Frameworks	Similarity with other Frameworks	Type of Data Mining Algorithms Implemented	Possible future research directions
Map reduce (2004) [1, 2]	Exhibits Data Parallelism like Sector-Sphere and Graph Parallelism like Distributed Graph-Lab	Basically used for implementing Mining Algorithms used for spatial analysis and Biological Pattern Analysis	Network bandwidth can be optimized and provision for user specific functions can be incorporated. Task parallelism can be thought of as a new dimension in this framework. The problem of heterogeneity can be addressed as a part of future research
Distributed Graph lab (2012) [11, 12]	Shared Memory as Map Reduce	Used for implementing mining algorithms that deals with 2D or higher dimensional data	The separation of storage and compute node can be incorporated for efficiency enhancement and implementation of dynamically evolving graphs can be researched
Parameter Sweeping (2011) [10]	Task Parallelism as Sector Sphere	Generally large-scale rule mining algorithms are implemented atop of this framework	Implementation of spatial and Machine-learning algorithms with task parallelism can be implemented with different parametric values
Sector Sphere (2008) [6]	Data Parallelism as Map Reduce and Task Parallelism as Parameter Sweeping	Implementation of mining algorithms concentrating on removing noisy data is a trait of this framework	Other major mining tasks can be implemented in this framework for betterment of data mining community
Spark (2009–10) [9]	Extension of Map Reduce and Hadoop having Shared Memory	Spark is used to implemented algorithms performing statistical and geo-spatial data analyzing algorithms	More mining algorithms with high space and time complexity can be researched with Spark to enhance their performance in terms of latency and turnaround time
Hadoop (2005) [3]	Higher version of Map Reduce displaying Shared Memory	Hadoop has been surveyed with supporting many machine-learning algorithms in mining as well as highly diversified data analytics	Can be experimented with text mining algorithms as well as intelligent analysis tasks
Pegasus (2009) [95]	Uses the underlying model of Hadoop and Map-reduce and support graph mining and parallelism as [11, 12, 96–102]	Optimized for web mining algorithms	Statistical mining models can be experimented
Airavat (2010) [59]	Map-reduce distributed file system storages are used as an underlying storage	Clustering and classification of big data	Rule mining algorithms can be experimented for security of user's data
FSBD (2014) [94]	Distributed architecture of Map reduce is used	Clustering of Big Data and business analytics	It is a potential future in SAAS respect and much algorithms needs to be implemented in this framework for better scheduling
Pregel (2010) [96]	Graph parallelism as Distributed Graph lab [11, 12] and graph mining framework as [95, 97–102]	Clustering algorithms for social network analysis	The computation workspace is limited to the size of RAM and this can be researched in future for improvements
Giraph (2015–16) [97–99]	Created atop Hadoop [3] and architecturally similar to Pregel [96] and HDFS used as in Hadoop and Map reduce [1, 2]	Clustering tasks	New algorithms can be implemented for social network analysis
GraphX (2013–14) [100–102]	Built atop of Spark [9]	Genetic Algorithm for data mining	Researches are going on to orient it in Relational databases
Mahout (2017) [130, 131]	Architecturally similar to Hadoop [3] and Map reduce [1, 2]	Versatile and many data mining tasks are supported	Researches should be conducted to provide out of core operational support

Table 6. Evaluation Matrix of Cloud Data Mining Frameworks

Cloud Data Mining Frameworks	HDFS/GFS/MapR supported	Sector Storage model	Shared Memory support	Fault tolerance	Scalability	Robustness	Reliability	Data Consistency	Security	Latency and Efficiency	Distribution
Map reduce (2004) [1, 2]	YES	NO	YES	YES	YES	NO	NO	NO	NO	Moderate	Moderate
Distributed Graph lab (2012) [11, 12]	HDFS Only	NO	YES	YES	YES	NO	NO	YES	NO	Moderate	YES
Parameter Sweeping (2011) [10]	DFS only	No	NO	NO	YES	Moderate	NO	NO	NO	Good	NO
Sector Sphere (2008) [6]	NO	YES	NO	NO	YES	YES	YES	NO	NO	Moderate	YES
Spark (2009–10) [9]	HDFS /MapR FS/GFS	NO	YES	YES	YES	YES	YES	NO	NO	Good	Moderate
Hadoop (2005) [3]	HDFS	NO	YES	NO	YES	YES	NO	NO	NO	Bad	Moderate
Pegasus (2009) [95]	DFS	NO	NO	YES	YES	YES	NO	YES	NO	Moderate	YES
Airavat (2010) [59]	DFS /MapR FS	NO	NO	NO	YES	YES	YES	YES	YES	Good	Moderate
FSBD (2014) [94]	DFS	NO	NO	YES	YES	Moderate	NO	YES	NO	Good	Moderate
Pregel (2010) [96]	NO	NO	NO	YES	YES	Moderate	YES	YES	NO	Good	YES
Giraph (2015–16) [97–99]	HDFS	NO	NO	YES	YES	Moderate	YES	YES	NO	Good	Moderate
GraphX (2013–14) [100–102]	NO	NO	NO	YES	YES	YES	YES	YES	NO	Good	Moderate
Mahout (2017) [130, 131]	NO	NO	NO	YES	YES	YES	YES	YES	NO	Good	Moderate

Cloud promises high level of distribution over its resources to attain scale up, speedup and better throughput. But, it is to be considered that high distribution leads to inconsistency if not addressed properly. Distributed Graph-Lab [11, 12] has achieved fair enough consistency with distribution. Data replication functions need to be robust to maintain maximum consistency between distributed data in cloud environment. In case of static data mining problems, data consistency can be omitted as parameter but if the mining is performed on dynamic and high velocity data, data consistency is very important. Frameworks must be capable of maintain data consistency across various sites if the data is distributed and duplicate copies are maintained. The cloud storage is a house of many storage nodes and distributed databases. Mining such databases is a problem as any changes in the data can make the database eventually consistent and it takes time for full consistence. Read Your Own Write (RYOW) is generally used to achieve consistency. This notion can be utilized in cloud data mining tasks over distributed datasets and databases [65–67]. Table 8 shows some of the major performance improvements implemented in cloud data mining frameworks and their limitations. From the scenario, it is seen that Map-reduce and Hadoop frameworks have been

Table 7. Type of Parallelism Supported in Cloud Data Mining Frameworks

Cloud Data Mining Frameworks	Data parallelism	Task parallelism	Graph parallelism and mining
Map reduce (2004) [1, 2]	YES	NO	YES
Distributed Graph lab (2012) [11, 12]	YES	NO	YES
Parameter Sweeping (2011) [10]	NO	YES	NO
Sector Sphere (2008) [6]	YES	YES	NO
Spark (2009-10) [9]	YES	NO	NO
Hadoop (2005) [3]	YES	NO	NO
Pegasus (2009) [95]	YES	NO	YES
Airavat (2010) [59]	YES	NO	NO
FSBD (2014) [94]	YES	NO	NO
Pregel (2010) [96]	YES	NO	YES
Giraph (2015–16) [97–99]	NO	NO	YES
GraphX (2013–14) [100–102]	YES	NO	YES
Mahout (2017) [130, 131]	YES	NO	NO

Table 8. Optimization Techniques Implemented in Cloud Data Mining Frameworks

Frameworks	Optimization type	Provided by	Limitations
Hadoop (2005) [3]	Data Locality [3]	HDFS (Storage)	Difficult to cope with distribution and scalability
Sector Sphere (2009) [6]	Data Communication [6, 70]	UDT (Protocol) over WAN	Distribution to remote servers is an issue
Map Reduce (2010) [1, 2]	Data Locality [72]	Map-reduce aware scheduling (Scheduling Technique)	Achieving scalability can cost effort and time
Map Reduce (2010) [1, 2]	Automatic Performance Tuning in parameters [74]	Cost-based Optimization	This optimization is framework-specific and non-versatile
Map Reduce (2011) [1, 2]	Automatic Performance Tuning in I/O-MANIMAL [75]	Analyzer Functions and Compression Function	Framework-specific and non-versatile
Map Reduce (2011) [1, 2]	Performance Tuning using Sample Data [76]	4 optimizing parameters used for tuning	Lesser options for users
Hadoop (2011) [3]	Starfish-Performance tuning [78]	Self Tuning Databases System [78]	Framework-specific system
Hadoop (2013) [3]	SVR performance Model [77]	Machine-learning Model	Framework-specific system

target for performance optimization and research needs to be extended towards other cloud data mining frameworks. Moreover, it needs to be researched if hybrid optimization solutions can be built for the frameworks combining the advantages of two or more options.

2.6 Some Discussions and References on Major Cloud Platforms and Services for Data Mining-related Tasks

Public Cloud services are getting popular day by day for their pay per use policies. And some of the services have well suited applications and resources for running Data Mining and Machine-learning-related tasks. While going through the literatures and researches, we have found that Big data Analytics are being carried out in some of the Cloud services. Reference [168] (which

has also been cited as related survey in Section 1) has some discussions on the various available Cloud services as Amazon AWS, Amazon EC2, Google Cloud and Microsoft Azure. Reference [177] cites Amazon EMR (Elastic Map-reduce) services and MLBase. The paper also states about Spark Streaming for real time high velocity streaming data processing. IBM Cloud Bluemix [181] is a PaaS facility in Cloud that supports Data Mining and Machine-learning languages such as Python and Java. Reference [182] discusses a new service atop Bluemix, which is dedicated for Mobile Cloud development applications. Google Machine-learning Engine [183] is a service of Cloud that enables practitioners and data analysts create highly scalable machine-learning and AI-related applications. General Electric's Predix [184] is a Data Analytics and Machine-learning platform for Industrial machines. It is an Edge-to-Cloud system that enables analytics and Learning from data in edge devices and analyze data from various devices by transferring them to Cloud using Scalable Predix framework. It also provides excellent applications for Data Visualizations after proper Analytics of Big data from IOT devices. The TCS TCUP (TCS Connected Universe Platform) [185] is an excellent scalable PaaS service over the Cloud that meets the three V's (Volume, Velocity, and Variety) of Big Data analytics efficiently. It can be used for efficient extraction, storage and analysis of data from various connected IOT devices and performs. This allows Machine Learning and Data Analytics on Streaming data or real time analytics. IBM's Watson [186] is an expert platform for Deep Analytics and AI applications for industries and enterprises. The main features of this PaaS is security of enterprise data and efficiency of applications. Microsoft Azure's ML (Machine-learning) platform [187] is an excellent platform consisting of many Microsoft services. It has capabilities for predictive analytics of real time streaming data. It has been already used in Anomaly Detection, Customer Churn Management and other such applications. Big data analytics languages such as R, can be directly used with Azure ML with ease. The user friendliness of this platform is its major attraction.

Velocity aspect of Big data is gaining great importance due to the highly dynamic data streams being generated from IOT devices and other Cloud environments. Data must be handled within very hard deadlines for high velocity data to extract meaningful information out of it. We have surveyed some services, platforms, and frameworks that support data mining of high velocity fast data. The surveys [191, 193] stated in Table 1 (in Section 1) are recent papers that discuss the velocity aspect of Big data in cloud environments. The survey [193] has placed some platforms on streaming data analytics. The famous out of them being Spark Streaming, StreamDM, MOU, Apache SAMOU, Storm, Apache S4, Samza, Aurora, Amazon Kinesis, Kafka Integrated SQLstream Blaze, and Pulser (all these platforms and services are cited appropiately in Reference [193]). As already stated in Section 2 under Spark framework, Spark Streaming [189] is an extension of Spark for handling high velocity data analytics. StreamDM [190] is an extension of Spark streaming for advanced fast data mining for streaming data. Massive Online Analysis (MOU) [192] is a framework for streaming data mining written in Java. It has excellent support for Clustering, Classification, and Regression algorithms. It is highly portable and dynamic but non-distributed in nature. It can handle concept drift (i.e., the over time change of statistical properties of parameters in predictive analysis) in a very impressive manner. Apache SAMAO (Scalable Advanced Massive Online Analysis) [194] is a fast distributed mining framework with library for Big data streams. It provides algorithmic support for Clustering, Classification, and Regression data mining tasks. SAMAO can be integrated with other distributed engines as Storm, S4, and Samza [193, 194]. We will mention and give brief details about streaming data analytic algorithms in Section 3. Figure 17 is a categorization of the surveyed Frameworks, Platforms, and Services on the basis of Parallelism approach (Task, Data, Graph parallelism) and Processing support (Batch and Streaming), which is the *Velocity* aspect in Big data.

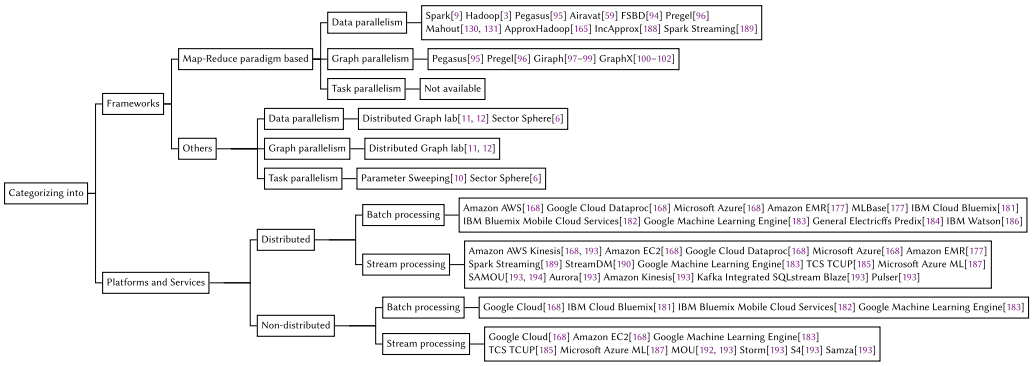

Fig. 17. The categorization of all the surveyed frameworks/services and platforms of big data analytics and machine learning on the basis of parallelism, distribution and streaming data processing support.

3 OVERVIEW OF ALGORITHMS IMPLEMENTED IN CLOUD DATA MINING FRAMEWORKS

Classical algorithms for data mining have really impacted a lot in the fields of trend analysis and pattern mining in various fields of science and technology. The classical algorithms can be categorized into some paradigms, the *Descriptive* paradigm, *Associative* paradigm, *Discriminant* paradigm, and *Predictive* paradigm. These paradigms generally fit into two learning models: *Supervised* and *Unsupervised* learning system. Clustering techniques fall under the *Descriptive* paradigm, which is also unsupervised. Association Rule mining comes under *Associative* paradigm, which can be both supervised as well as unsupervised. Classification falls under *Discriminant* paradigm, which is supervised and Regression comes under *Predictive* paradigm, which is purely supervised in nature. These paradigms are highly motivated by discovery models and are implemented in cloud frameworks except Regression, which is yet to be implemented in cloud environment. The verification model base techniques are yet not implemented in cloud frameworks and can be considered for future research area. The below given tree in Figure 18 has been placed to give a clear picture of the above mentioned categorization. Table 9 gives a classification of Cloud Data mining approaches by grouping them into Clustering, Classification and Association rule mining techniques.

3.1 Clustering and Pattern Discovery

Clustering is the process of grouping data into groups where similarity between data is maximized and dissimilarity is minimized. Here the groups are not known previously and number of groups to be formed is also not surely known. Clustering is generally unsupervised in nature and no training data is fed into the system for modeling the system. The major categorizations of clustering techniques implemented in cloud are: *Hierarchical*, *Partitioning*, *Bi-Clustering*, *Soft Clustering* based and *Density* based. The Hierarchical-Agglomerative [15] algorithm and CURE [23, 24] falls under Hierarchical technique. Distributed K-means [14, 21, 116–118, 119] is a partitioning-based algorithm. DB-SCAN [82] is a density-based approach and variants to this algorithm implemented in cloud are Spark-DBSCAN [33], NG-DBSCAN [31], MR.SCAN [139], RP-DBSCAN [196], and MR-DBSCAN [32]. Some specific approaches are Bi-Clustering [16], Fuzzy Clustering [22], and Genetic algorithms. Bi-Clustering has two algorithms implemented in cloud: BiTM-MR [16] and DisCo [18]. FCM [22] is an algorithm implemented with fuzzy approach. And, a hybrid technique Genetic K-Means [30] has been implemented with the idea of distributed K-means [10, 21, 28] and genetic algorithms. Figure 19 provided below shows the above stated categorization clearly.

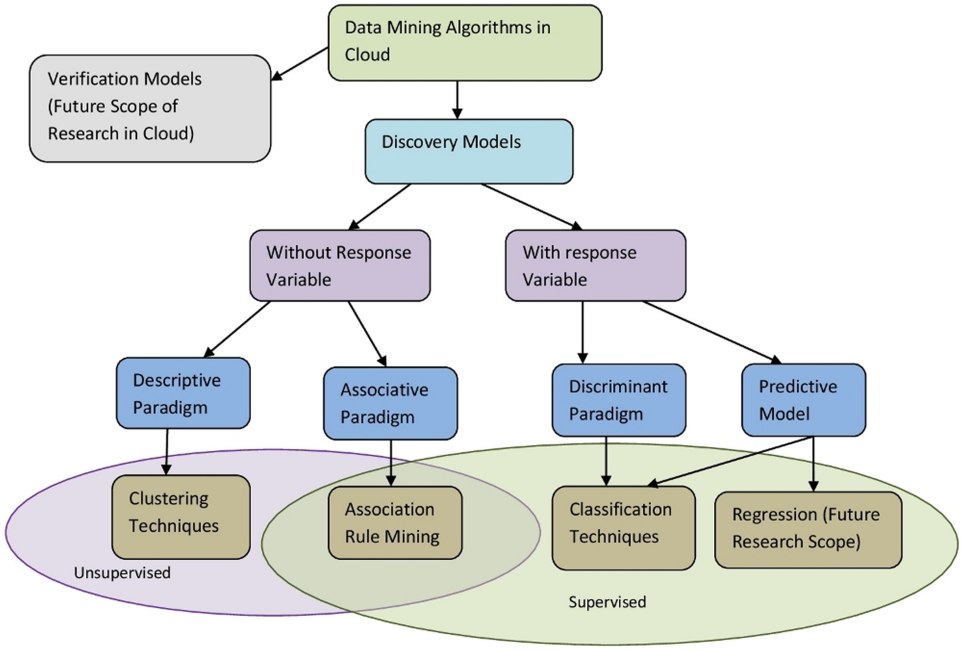

Fig. 18. Categorization of data mining techniques implemented in cloud.

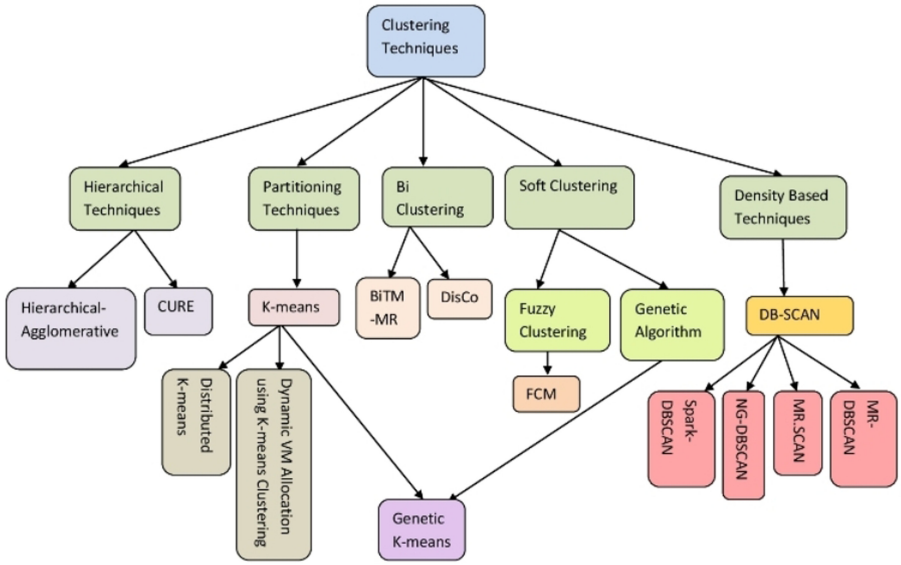

Fig. 19. Taxonomy of clustering techniques implemented in cloud.

Cloud Data Clustering is a slightly higher concept in data mining. The normal clustering is fused with the real power of cloud services in this concept. The facilities such as parallelism, data availability, reliability and security are attached with cloud data clustering.

3.1.1 Hierarchical Clustering Techniques. Hierarchical Clustering is a type of clustering where the clusters are formed using hierarchical methods. The data is grouped according to some

Table 9. A Classification Based on Different Cloud Data Mining Approaches

Classification	References
Clustering Techniques:	
Hierarchical Clustering	Hierarchical Agglomerative Clustering [15], CURE in Cloud [24], Hierarchical Clustering in Cloud [19], Agglomerative Clustering in Cloud [20], Hierarchical Clustering in cloud [25], ClusTree [198]
Partitioning Approach	K-Means Clustering [10, 14, 28, 116–119], Distributed K-Means Clustering [21], Dynamic VM allocation using K-Means [29], Parallel K-Means [26], CluStream [197], CluStream [194]
Density-based Clustering	MR-DBSCAN [32], Spark-DBSCAN [33], MR.SCAN [139], NG-DBSCAN [31], RP-DBSCAN [196], DenStream and D-Stream [199, 200]
Bi-Clustering Approach	Bi-Clustering BiTM-MR [16], Parallel Bi-Clustering [17], DisCo [18]
Soft Clustering	Web FCM [22], Parallel Genetic Algorithm [27, 81]
Hybrid Clustering Approach	Genetic K-means [30]
Classification Techniques:	
K-NN-based Classification	PPKC (Privacy Preservation KNN-based Classification) [38], KNN-based Classification [39], Parallel K-NN [180]
Neural Network	Neural Network-based Classification [83]
Bayesian Classifier	Naive Bayes parallel Classification [36], Parallel Naive Bayesian Model [180]
Support Vector Machine	Parallel SVM [34]
Decision Tree-based Classification	Parallelized SPRINT [35], Parallel Decision Tree-based Classification [180], EVFDT [201], Class-based micro-classifier ensemble classification technique (MCE) [202], A parallel random forest algorithm for big data in a spark cloud computing environment [203]
Genetic Algorithm-based Classification	Genetic Algorithm for Cloud [37]
Point data-based Classification	Point data Classification in Cloud [41]
Hybrid Classification Approach	Naive Bayes and SVM-based Classification [40]
Association Rule Mining Techniques:	
FP-Growth Tree-based Approach	PLFPG [48], EPF [54]
Apriori-based Approach	Parallel implementation of Apriori [90, 91], Apriori in Cloud [43], Improved Apriori Algorithm (CBPA) [47], MR-Apriori [42, 179]
Frequent Pattern Mining Approach	Frequent Pattern Mining in Cloud [44, 45], PaMpa-HD [53]
Other Approaches	SEARUM [51], FMAAR [55], HUDS [52], Association rule mining in Cloud [49, 50, 56]

similarity in a hierarchical manner. Two well-known approaches in this regard are Bottom-up and Top-down methods. The clustering is generally performed either by splitting the entire data set according to some dissimilarity parameter or fusing the datasets according to some similarity measure hierarchically. Some algorithms have been identified in this category, which has been implemented in cloud environment.

A distributed cloud computing environment is used to implement a hierarchical agglomerative clustering algorithm. Normal Hierarchical Clustering is suited for small datasets and in the case of large datasets, such kind of algorithms can consume a lot of time as the data set size increases. So,

to facilitate large datasets using hierarchical agglomerative clustering, Kriti Srivastava et al. [15] introduced the parallel version of the algorithm in cloud framework. The working principle of the algorithm is quite simple; the whole dataset is divided into some subsets. The clustering algorithm can be applied in the individual datasets to produce local clustering results. The local results are combined with the previous results and the steps are followed iteratively to form the main cluster. This approach is a combination of two stages: Micro Clustering and Macro clustering. The main tasks containing in this approach are: Applying Kmean algorithm on layer 1, which consists of some partitioned and distributed datasets derived from various sites; second, merging of files with k centroids and the corresponding clusters into one master file; and finally, applying the hierarchical Agglomerative clustering approach in the third layer and generating output in the form of dendogram. The efficiency of this technique has been established in cloud-based framework.

Madhuri h Parekh et al. [24] have further suggested a new algorithmic paradigm for cloud in agglomerative approach using CURE hierarchical algorithm. Use of parallelism in algorithm increases the efficiency and improves the latency of the system. The Map-reduce framework has been used in this respect to parallelize the system and divide the data into sub parts so that the retrieval from cloud storage can be availed. The basic idea of integrating CURE algorithm in cloud method is due to the advantages of CURE in real life data processing [23]. The CURE is capable of identifying non-spherical clusters in the case of spatial data analysis and is very robust in the case of outlier detection and exclusion from clusters. Multi-sized clusters can be detected very well by this algorithm. Now, Hadoop Map Reduce has the capacity of high scalability and Availability over much complex data in structured as well as unstructured format. Now comparison has been made by the researcher about centroid linkage hierarchical clustering with CURE in cloud framework and found that the second one is much more improved in the case of performance.

Managing data in cloud can be a very nontrivial job. The type of data being uploaded into cloud by various end users is of different variety. To categorize and manage them for use, it is required to cluster the data based on some parameters or features. Esha Sarkar et al. [19] have proposed a technique to accomplish the task. Hierarchical clustering approach is been employed to cloud unstructured data from various sources such as educational institutions, hospitals, government organizations and private agencies. The algorithm is capable of forming clusters out of the whole data and it categorizes the data into groups with respect to the sources they come from. The advantage of this approach in cloud mining is that it is fast enough and provides the usage information of cloud storage space. The scalability is also appreciable and efficiency of clustering on heterogeneous data is also tested.

Madhuri h Parekh et al. [20] introduced an extension for agglomerative clustering algorithm for cloud data mining. The distribution of data over heterogeneous network is applied. In this approach heterogeneous networks are used in applying distribution of agglomerative hierarchical technique on data. The data are initially considered as one cluster. The similarity and dissimilarity parameters are calculated between each pair of objects and are merged into one cluster according to their degree of similarity or considered as noise in accordance with their degree of dissimilarity. A binary hierarchical cluster tree is used for the grouping of the data entities over the cloud.

Kun Qin et al. [25] have proposed a new spatial data mining method for cloud framework using hierarchical clustering. FY2C images are used as weather sampling spatial data, clustering is carried out upon brightness values of five wave bands by hierarchical approach. The varied weather types are processed by generating parameters based on image features. Cloud generator function is used to calculate the membership of each pixel to a cluster in cloud framework. Finally it will determine the precipitation weather type. The Kosko subset criteria are used to achieve multi level clustering in cloud. The experimental analysis of this technique has proved that it is capable of distinguishing rainy and non-rainy weathers but not so effective for other weathers.

3.1.2 Partitioning Clustering Techniques. This type of clustering jobs are accomplished by partitioning the data set into groups of data elements with respect to some predefined parameter for the number of clusters to be formed. Basically statistical calculation of data in quantitative manner using mean and median is used. Some algorithms implemented in the cloud environment are as follows:

A. Mahendiran et al. [14] have proposed an implementation of Kmeans algorithm in cloud environment. The algorithm has been implemented in Cloud SQL in Google cloud. The experimental results for this implementation have shown that it is capable of storing and analyzing huge amount of data and forms clusters with high accuracy. The algorithm uses the Google App Engine framework for its efficient implementation. A very high degree of parallelism can be achieved in Kmeans algorithm in cloud using cloud data clustering framework. Task as well as data parallel implementation can further be improvised. If we need to form clusters in Kmeans with $K = 4$ and $K = 6$, then it can be achieved using task parallel approach of parameter sweeping framework. Also huge storage of data can be availed in cloud framework in less cost.

Further, Renu Ansari [21] has suggested a new distributed k-Means clustering algorithm for cloud data in her paper. Cloud is a collection of significantly unstructured data and it is a challenge to process them rightly and accurately. The method discussed is especially applicable to huge volume text mining in big data storage. The technique is build upon the notion that the texts are oriented in accordance with the social networking data. The user mood is an important case to be considered in such cases of text analysis in mining. For experimentation purpose Twitter data has been used by the researcher. The algorithm considered is K-Means algorithm for finding the hidden sentiments of text writer. The algorithm has been developed in JAVA and to utilize it in cloud as a service the Open Shift environment is chosen. The algorithm is evaluated with many parameters such as accuracy, error occurrence, time and space complexity and it is proved to be highly performing for the said contexts. Further discussing the infrastructure and technology, Hadoop storage is used to store the data to be processed. Data pre-processing step is also carried out to remove noisy data and unwanted data from datasets. The NLP tool is used in tagging the data with features hence used in feature extraction.

K-Means clustering algorithm has been modified with the notion of approximate nearest-neighbor search by Michael Shinder et al. [28] It is reported to be faster and efficient for huge datasets. The conventional Divide and Conquer notion has been replaced with the newer notion and it has yielded better results. The Cloud distributed framework is an ideal area for this algorithm to be executed in Big Data.

Bhupendra Panchal et al. [29] proposed a dynamic virtual machine allocation algorithm using Clustering in Cloud framework. The proposed technique uses K-Means clustering method for forming clusters. It is implemented in Cloud Sim. It has shown promising results in data-centers. The quality of this technique can be shown in terms of load sharing and efficient CPU utilization.

Spectral clustering has gained importance recently due to its high performance than normal clustering. The problem with spectral clustering is that it is highly dependent on complex computational methodology. Ran Jin et al. have come up with a technique that is much faster than traditional spectral clustering using cloud framework. Sparse matrix Eigen values are calculated in distributed clustering mode. With the increase of volume in processing data the rate of clustering is linearly increasing. The K-Means algorithm is used in parallel mode in Map-reduce framework. The main complexity is concentrated in construction of matrix and finding the feature vectors in Laplace matrix. The matrix construction and feature vectors selection are done in parallel along with K-means clustering using the cloud framework to achieve the desired speed up in spite of computational complexity of spectral-based paradigms [26].

3.1.3 Density-based Clustering Techniques. Density-based clustering techniques are generally based on density functions. The clusters are formed by evaluation of the density for a region in space. The closely located data items are grouped together into one cluster and the sparsely located data items are grouped into another cluster and so on. This technique is famous for finding multi density and nester clusters and patterns in data sets. Some parallel versions of the pioneering DBSCAN [82] algorithm in density-based technique has been implemented over the cloud.

Many Parallel implementations of DB-SCAN density-based clustering algorithm has been proposed in recent times (MR-DBSCAN [32], Spark-DBSCAN [33], MR.SCAN [139], and RP-DBSCAN [196]). These implementations have been done in Map-reduce and Spark Frameworks. However, the recent researches by Alessandro Lulli et al. have shown the possibility of parallel implementation of DB-SCAN in Graph-Lab framework-NG-DBSCAN [31]. It can operate on any arbitrary Cloud Data and symmetric distance functions. It is capable of producing clusters with high quality very quickly. Vertex centric computing is incorporated here. The data is partitioned as per computation in a vertex of a graph locally. The input data can be taken from adjacent vertexes and the output can be sent through the outgoing edges. The algorithmic implementation has shown promising performance in 2D-Datasets and text mining systems.

Yaobin He et al. [32] use Map-reduce framework to implement a parallel density-based clustering technique. DBSCAN algorithm of density-based notion has been considered for the case. The implementation is carried out in four stages. The data is preprocessed and cost-based partitioning is carried out, then local DBSCAN carried out in nodes, then the merge mapping are found and finally the merging of local results is done to give the final global results. The data partitioning is done with non-indexed spatial data. This algorithm is optimized for large data I/O. It works with minimum spatial and computational complexity. Proper speedup and scale up has been achieved.

Dianwei Han et al. [33] have proposed a parallel implementation of DBSCAN algorithm using Spark. Kd-tree has been used for searching. Cost-based partitioning is done. The data structures have also being selected as per the requirement of data processing. Java has been used as the underlying platform for implementation of this technique. The speedup and scale up are found to be quite impressive.

Random Partitioning-DBSCAN (RP-DBSCAN) [196] proposed by Hwanjun Song et al. uses random partitioning approach. It does local clustering on data partitions and finally merges these clusters to one global cluster. It has been implemented atop Spark framework using Microsoft Azure services. The algorithm works excellently well compared to its counterparts (180 times better) due to its lightness in data partitioning and merging processes.

3.1.4 Bi-Clustering Techniques. Bi-clustering is a very famous technique introduced in 1972. It is a specialized technique used in producing subset of data in matrices, where we can consider column data to determine rows exhibiting similar behavior or row data to determine columns exhibiting similar behavior. Based on such behaviors, the data is grouped into subsets. Constant as well as coherent values can be considered over columns and rows. It is basically a form of normal clustering where we can produce bi-clusters having rows or columns that have similarity in one way or another with respect to some subset of columns or rows in the same matrices.

Some works have been accomplished in the field of parallel or distributed bi-clustering. The framework utilized for this algorithm to be implemented in distributed parallel environment is Map Reduce as introduced by Tugdual Sarazin et al. [16]. The BiTM-MR algorithm in bi-clustering has been successfully implemented in Map-reduce Framework using Spark. The distributed implementation of this algorithm using Spark concentrates on failure correction, high distribution and management of data. The main attraction of such a technique is to make task parallelism in the function of row and column iteration. Self organizing maps as in unsupervised learning techniques have been used in the above algorithmic implementation.

Gene Expression data clustering has been a very important task in the field of data mining in bioinformatics. The bi-clustering technique of data clustering is widely used to recognize meaningful information and patterns in genes. Liu et al. [17] proposed a parallel implementation of a bi-clustering algorithm to speed up things and efficiency building in evaluation of the rows and columns contained in the gene matrix. The final bi-clusters are obtained by adding rows and columns to a subset of the matrix of genes incrementally. The efficiency of this approach is found to be superior to other non-parallel approach and the quality of clusters is also good. The framework to be used for this algorithm to fit in cloud can be Graph-Lab. Although it has yet not been implemented in cloud environment but it can be a scope of future work to go with graph-lab for this algorithm.

Map Reduce has been a very famous framework for bi-clustering or co-clustering application in large-scale data mining. A distributed co-clustering algorithm (DisCo) was introduced by Spiros Papadimitriou et al. [18] to process huge data sets using Map-reduce framework. This data preprocessing and co- clustering algorithm is being implemented with Hadoop Map-reduce framework. It can scale well in the case of thousands of Gigabytes of data maintaining its efficiency and robustness.

3.1.5 Soft Clustering. Broadly speaking, clustering can be grouped as hard and soft clustering techniques. Hard clustering means the corresponding data belongs to only one group according to some feature. But this is sometimes very unpredictable as the data might exhibit multiple features across different groups. This is where soft clustering comes to light. It is the technique where data elements are assigned to more than one cluster as per the features they display. Soft clustering techniques implemented in cloud are generally fuzzy based and genetic algorithmic based. Fuzzy-based techniques are based on membership of data items in more than one cluster. Genetic clustering mimics the process of biological evolution of our body.

Xianfeng Yang et al. [22] introduced a web fuzzy clustering model for high volume data in cloud environment. The author has used a web object similarity matrix or a web fuzzy equivalent matrix. The performance of the overall data mining problem has been improved in cloud using this fuzzy technique. The algorithm is implemented with a user's interface for providing user services. The data is first cleaned for any noise using Call Services as a method of data preparation for mining. After this step in data layer, the Call Interface stage puts the data in the cloud storage for the actual mathematical fuzzy method to process the data for mining results. FCM can be integrated with cloud framework to implement efficient fuzzy-based clustering algorithms.

Genetic algorithms are a class of algorithmic paradigms that mimic the natural selection process based on biological evolution. These set of algorithms are used to solve optimization problems following an intelligent heuristics. In the advent of producing more and more efficient cloud-based mining systems Nivranshu Hans et al. [27] have proposed a big data clustering using Genetic Algorithms. The author has stated that Map-reduce framework has been used to realize this technique. Parallelism in the case of Genetic Algorithms (GA) can be achieved using two of the models Coarse Grained or fine Grained parallel GA. The Coarse-based technique is used in parallelizing GA in cloud. The Mapper function is fed with data sets split as per the size of blocks. The Mapper does the local clustering and the result of Mappers is sent to one reducer node to perform the second phase of clustering globally.

3.1.6 Hybrid Clustering Techniques. Hybrid techniques in clustering are the methods where two or more approaches are fused to create new robust application specific approaches for data mining.

Pooja Bisht et al. [30] have proposed a hybrid technique for clustering Big data in cloud environment using Genetic paradigm in k-Means approach. The algorithm combines the features of

Genetic Algorithmic Paradigm of heuristics and excellent statistical model of K-Means Clustering to provide an optimal solution for Big Data Mining. The main concern here is the time complexity. Though the parallel implementation of this technique is not mentioned in this article but extensive study on this technique shows that it can be successfully implemented in Cloud Map-reduce or Sector Sphere framework for attaining Parallelism and good turnaround time. Table 10 makes a comparison of various Clustering algorithms implemented in Cloud environment discussed in the above section. The table also discusses the three V's (Volume, Variety and Velocity) of big data for each of the algorithms by mentioning about the parallelization approach, variety of managed data, scalability and streaming data processing capability.

The above categorization of algorithms have mention of high volume and variety data clustering algorithms. We have also identified some algorithms in clustering specifically for streaming data mining in cloud environments. In Partitioning method, CluStream [193, 194, 197] is prominent, Hierarchical method being ClusTree [193, 198] and Density-based technique covers DenStream and D-Stream [193, 199, 200]. Some more techniques are cited in Reference [193]. Reference [195] has placed proper categorization of Density-based streaming clustering algorithms. It has been classified into micro clustering and grid-based clustering and places the algorithms accordingly.

3.2 Classification and Prediction

Classification is the technique of assigning data items or objects to classes or to predict data trends. Models can be formed as per user's requirement to categorize data. Classification is a term related to categorical data prediction. The challenge comes in developing scalable classification techniques that will be capable of processing large data. Some of the famous approaches in classification process are *decision tree technique*, *Bayesian Classifier*, *Support Vector Machines*, *Neural Networks*, *Rule-based classifiers and belief networks*. The cloud data mining frameworks have realized some classification techniques as K-NN Classification [38, 39] and Point data classification [41]. Naive Bayes parallel technique [36] comes under Bayesian Classifier. Parallel SVM [34] for cloud is a Support Vector Machine-based technique. Parallelized SPRINT [35] is a Decision tree-based technique. Genetic Algorithmic Classification [37] has been successfully realized in cloud. A hybrid technique for data classification is based on Bayesian Classifier and SVM [40]. Reference [180] has experimented with parallel versions of K- Nearest Neighbor, Naive Bayesian model, and Decision Tree atop Hadoop Map-reduce framework, which resulted in good accuracy in Big data analytics, linear scalability and high parallelism. Figure 20 shows a clear picture of various Classification techniques implemented in Cloud.

3.2.1 K-Nearest Neighbor (K-NN)-based Classification. K-NN-based classification is a type of classification method where classification of a data object is based on the votes of its neighbors. The data object becomes the part of the class that is most common in its neighborhood. The neighborhood has k neighbors to assume. Some K-NN-based classification techniques have been implemented in cloud environment. Pooja Bajare et al. [39] have further proposed another technique for classifying data into useful buckets without disclosing the data to vulnerable people in cloud. The users now a days prefer to host encrypted data in cloud to preserve data privacy of the user. The problem here encountered is that algorithms are generally not designed to perform mining and classification on encrypted data so the researchers have placed a KNN-based classification algorithm to work on encrypted data. Elliptic Curve cryptography has been used instead of SMC-Secure Multi-party Computation. The baseline of this technique is that: Intermediate results and output of classification tasks are not disclosed to cloud. User queries are not disclosed to the cloud environment. The result of the tasks must only be conveyed to the user and none other. No other information other than result set must be sent back or disclosed to users. Hence, security of data is

Table 10. Comparison of Clustering Algorithms Implemented in Cloud in Chronological Order

Algorithms and Citations	Year of implementation in Cloud	Cloud Framework used	Optimized for	Advantages	Noise Handling	Parallelization approach	Types of data managed	Scalability
DisCo [18] (Citations-265)	2008	Hadoop Map reduce Framework	Large-scale Data mining	It can scale well in the case of thousands of Gigabytes of data maintaining its efficiency and robustness	Good	Data and Graph parallelism	Un-structured and Semi-structured data	High (Terabytes of data), scales linearly with number of machines/nodes
K-Means [116] (Citations-587)	2009	Hadoop Map reduce	Huge datasets	Capable of processing huge data with good scalability and speedup	Moderate	Data parallelism	Un-structured	Scales well with Gigabyte scale data
K-Means [117] (Citations-75)	2011	Apache Mahout atop Hadoop	Performance gain	Good Consistency, scalability and speedup gains	Moderate	Data parallelism	Un-structured data	High (Gigabytes of data, 5 million records, 41 attributes)
MR-DBSCAN [32] (Citations-139)	2011	Map-reduce Framework	Minimizing frequency Large Data I/O	Minimum Spatial and Computational Complexity, Speedup and Scale up is good	Good	Data parallelism	Structured data	High (1.9 billion GPS location records)
K-Means Clustering [28] (Citations-122)	2011	Cloud Distributed Framework	Huge streaming Data	Very efficient with huge data	Good	Data stream parallelism	Structured	High (11.6 million data points with 57 integer features and 2,458,285 points in 68 dimensions)
K-Means [14] (Citations-26)	2012	Google Cloud Using Google App Engine. Hadoop Map reduce	Huge datasets and real time data	Capable of storing and analyzing huge amount of data and forms clusters with high accuracy	Moderate	Not parallelized	Structured and Un-structured	Medium
K-Means [10] (Citations-26)	2012	Parameter Sweeping (Microsoft windows Azure service)	Massive Data and Task Parallelism	Effective use of cloud services and efficient mining	Moderate	Task parallelism (parameter value sweeping)	Structured	High (2,458,000 instances dataset)
K-Means [119] (Citations-15)	2012	Map reduce	Massive data	High performance and Scalability	Moderate	Data parallelism	Un-structured data	High
Hierarchical Agglomerative Clustering [15] (Citations-15)	2013	Distributed Cloud Computing Environment	Data Parallelism	High degree of parallelism is attained with small as well as large datasets/supports micro-macro clustering	Moderate	Data parallel execution in nodes	Structured	Linear Scalability

(Continued)

Table 10. Continued

Algorithms and Citations	Year of implementation in Cloud	Cloud Framework used	Optimized for	Advantages	Noise Handling	Parallelization approach	Types of data managed	Scalability
Web FCM [22] (Citations-3)	2013	Generalized Cloud Distributed Framework	Statistical Data	Highly Parallelized	Excellent	Data parallelism	Structured, Semi-structured and Un-structured	Medium
Dynamic VM Allocation using KMeans Clustering [29] (Citations-3)	2013	CloudSim Environment	Dynamic virtual machine allocation	Efficient CPU Utilization and Load Sharing	Not Known	Not parallelized	NA	High
CluStream [194] (Citations-7)	2013	SAMAO Platform	Streaming data mining	Efficiency and scalability	Good	Data parallelism	Structured	High
K-Means [118] (Citations-79)	2014	Map reduce	High performance	Efficient, robust and scalable	Moderate	Data parallel approach	Un-structured and semi-structured data	High, 2 to 4 billion data points and many dimensions
Bi-Clustering BiTM-MR [16] (Citations-2)	2014	Map Reduce-Spark Framework	Task parallelism in Row and Column Iteration	Failure correction, high distribution and management of data	Good	Data parallel	Structured and Semi-structured	High (1 million and 2 million observations with 20 and 40 features each)
Parallel Genetic Algorithm (GA) [27] (Citations-11)	2015	Map Reduce Framework, Apache Hadoop	Problem Solving using Intelligent Heuristics	Parallelism is attained with accuracy	Good	Input Data parallelism	Structured	High (169308 instances and 2 dimensions)
Parallel Genetic Algorithm (GA) [81] (Citations-12)	2015	Map Reduce Framework, Apache Hadoop (Amazon EC2 cluster)	Machine-learning tasks such as feature selection	High parallelism is achieved	Good	Data parallelism	Structured	Medium
Distributed K-Means Clustering [21] (Citations-2)	2015	Hadoop Storage HDFC is used in Cloud Environment	Large datasets	It can work well with significantly unstructured data	Good	Data parallelism	Unstructured data	Medium
CURE in Cloud [23, 24] (Citations-3295, NA)	2015	Map Reduce Framework	Noise Handling and Outliers Exclusion	Capacity of high scalability and Availability over much complex data in structured as well as unstructured format	Excellent	Input data parallely processed	Structured	High

(Continued)

Table 10. Continued

Algorithms and Citations	Year of implementation in Cloud	Cloud Framework used	Optimized for	Advantages	Noise Handling	Parallelization approach	Types of data managed	Scalability
DenStream and D-Stream [199, 200] (Citations-5,126)	2015	General Cloud environment with MongoDB Platform and Google Cloud Services	Streaming data clustering for Anomaly Detection	High distribution and parallelism	Good	Data stream parallelism	Structured and Semi-structured	High
SPARK- DBSCAN [33] (Citations-9)	2016	Spark	Minimizing noise in spatial data mining	Scale up, speed up	Excellent	Data parallelism	Structured and Un-structured	High (1 million high-dimensional points, 8 to 512 cores used)
NG-DBSCAN [31] (Citations-14)	2016	Distributed Graph-Lab	2D-Datasets	Efficiency and versatility	Good	Data and Graph Parallelism	Un-structured data generally (Spam emails- 4 million, Twitter- 5 million tweets)	High (with dataset growth and number of nodes in cluster)
Genetic K-Means [30] (Citations-3)	2016(only for Big Data)	Proposed Framework- Map Reduce or Sector Sphere	Cloud Mining Framework	Accuracy of Clustering Big Data	Good	Not parallelized	Structured and Semi-structured	Medium
CluStream [197] (Citations-4)	2016	Apache Spark	Streaming data mining	Accuracy of Clustering Big Data with good scalability and accuracy	Good	Data parallelism	Structured	High (very good speedup up to 4X with 100 dimensions, 10000 points and around 40 processors)
RP-DBSCAN [196] (Citations-NA)	2018	Apache Spark	Huge amount of skewed data	Accuracy of Clustering Big Data with good scalability and efficiency	Good	Data and graph parallelism	Structured	High (362 GB of data managed, nodes varying from 5 to 40)
CluSTree [198] (Citations-NA)	2018	MAO Platform and Google Cloud Services	Streaming data mining	Efficiency and Accuracy	Good	Not parallelized	Structured	Medium

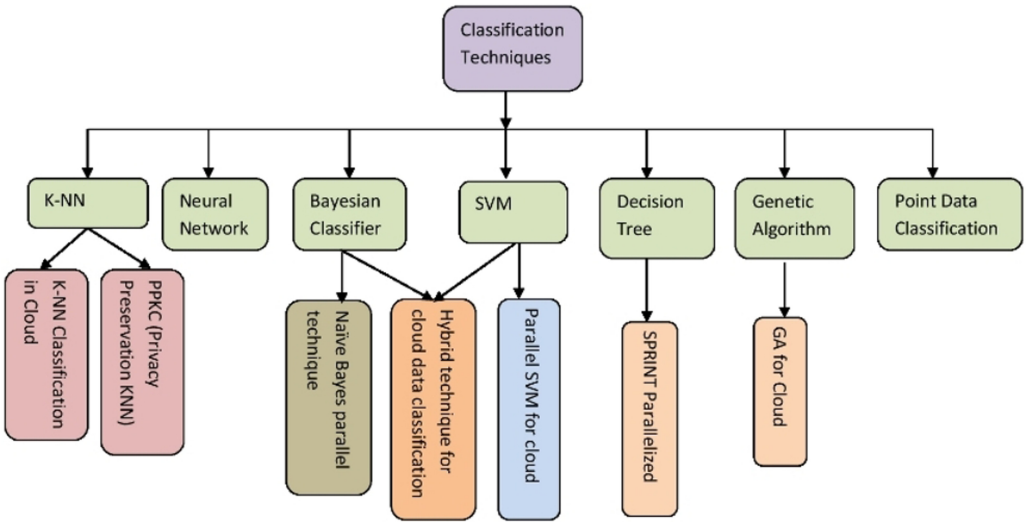

Fig. 20. Taxonomy of classification techniques implemented in cloud.

an important advantage of this technique. The full power of cloud computing has been leveraged by the organizations from recent times. The data is being hosted in the cloud and used for various processing using mining and segmentation techniques. And much success has been achieved in this field after the innovations in mining and processing frameworks such as Map Reduce, Sector sphere, Parameter Sweeping, and Graph lab.

The data is also exposed to the third party people in cloud and this is a concern in privacy and data vulnerability in business. The disclosure of sensitive business data is a great concern. Less research has been conducted on privacy of data in classification of data over cloud.

Jian Wang [38] has introduced a novel approach for maintaining proper privacy while mining data in cloud. PPKC (Privacy Preservation KNN-based Classification) is K-Nearest Neighbor-based classification technique used by the researcher for solving the purpose. The main concern of a privacy preservation algorithm is security of sensitive business data and, second, accuracy and efficiency. Now the classifier is made more efficient using Binary Weighted Cosine (BWC) metric measure for measuring similarity of records. Private Matching Protocol is used to protect the privacy of data without revealing private information while checking values from records for similarity. The output of BWC will be fed into weighted KNN for classification. The paper introduces data parallelism by doing local classification of subsets of data. Since, KNN is incorporated in the technique, so data parallelism is used here.

3.2.2 Neural Network (NN)-based Classification. Neural network is a paradigm that is based on human brain. It is a network of connected components called artificial neurons. These networks are capable of taking decisions on computations based on weights assigned to the components.

Neural networks are efficiently used in cloud environment to classify data. Medical data classification and grouping for analysis is done on cloud for better access and efficiency. A. Maithili et al. [83] have researched about neural network-based cloud data classification techniques for cancer data analysis.

3.2.3 Bayesian Classifier-based Classification. Bayesian Classification techniques are probabilistic techniques based on Bayes' theorem. The probability of a given data belonging to a class is predicted and class memberships are defined through probability.

Lijuan Zhou et al. [36] further put forward a new technique based on Naive Bayes classification. The Naive Bayes classification is based on statistics and probability. It has shown high perfection of data classification in limited latency. This probabilistic algorithm is merged with Map-reduce framework to achieve high parallelism in cloud. Hadoop framework with HDFS has been selected by the researcher to implement Naive Bayes' parallel technique. The technique is widely implemented in two steps: Training Stage and Prediction Stage. Since training and prediction are mostly dependent on the volume of data, the authors used the data parallelism technique here. The technique is proved to have better efficiency and scalability in parallel mode.

3.2.4 Support Vector Machine (SVM)-based Classification. Support Vector Machines (SVM) is a model capable of classifying data objects as per some supervised example data in a liner as well as non-linear manner. SVM-based classification is based on training data classification. The examples are assigned to one category or another on the bases of features. New data are classified into one or more classes or categories.

This technique has become famous in the recent times for its capability of classifying data in data mining as per customized user needs. Ozgur Catak et al. [34] have implemented the serial SVM in cloud environment with parallelism. Here, the famous Map-reduce Framework is made to use in implementation of training of classifier SVM in cloud. Training is related to voluminous data therefore data parallelism is employed in the training stage itself. Training of the data, which is distributed above cloud storage, is done in parallel. The support vector output in each compute node is collected and merged. This activity is carried out till optimal solution in SVM classifier is reached. This approach has achieved excellent scalability and concurrency of data set training in high level of parallelism.

3.2.5 Decision Tree-based Classification. Decision tree is a structure made of nodes and branches. Each node tells about a test performed on any parameter and each branch from that node says about its result. The terminal nodes in the structure tell about the label assigned to a class.

Lijuan Zhang et al. [35] have proposed a strategy to mine data in cloud environment using classification. The SPRINT algorithm is targeted to be implemented in cloud. SPRINT has been chosen by the researcher as it is a scalable and highly parallelizable model for data mining that can be run on multiple nodes for huge datasets. SPRINT is a decision tree-based classifier having a growth phase and a prune phase. Serial SPRINT technique has been implemented atop Map-reduce framework to convert it into parallelized SPRINT algorithm Coming to the mechanism of this algorithm in cloud, all the data in the training data set is submitted to different compute nodes. The nodes independently process the attribute list and partitions the data till the partitions cannot be divided any more either due to its high degree of similarity in contained data or due to is smaller size. Hadoop was used along with Ubuntu and eclipse in this experiment.

3.2.6 Genetic Algorithm-based Classification. Some classifications are inspired by biological systems of our body and natural processes dwelling in our body. These classification techniques comes under Genetic algorithm-based classifications. These are bio-inspired algorithms capable of producing better output through mutation, crossover and selection processes.

Jing Ding et al. [37] have proposed an approach in classification data mining using Genetic Algorithm framework over the cloud. The framework used is general master-slave framework having HDFS- or GFS-based storage. The Map-reduce framework has been employed attained maximum efficiency and accuracy. Crossover and Mutation are the key features in these techniques. The authors have exploited task parallelism here. Subsets of data are handled individually using task level parallelism. Data parallelism may not be that beneficial in this case as it uses genetic algorithm

for classification. A vast expense of global population is generated for the said operation and the possibility of local search is mitigated for better throughput. Randomized probability-based adjustment of algorithm is carried out mitigating manual effort. The fitness functions are dedicated to the user specific goal of mining jobs and allow better accuracy.

3.2.7 Point Data-based Classification. Spatial data analysis and image-processing techniques require classification of point data into appropriate categories. Point data, spatial data or pixel data in the case of raw images needs to be classified in parallel for better throughput and speedup.

Kun Liu et al. [41] have proposed a technique for point data classification in cloud framework. The point data processing has gained importance in geospatial analysis and many more. Apache Spark is used to implement this technique. A supervised learning classification technique is used comprising of feature computation, model training and prediction. The model was parallelized in Apache Spark 1.4.0 with a master node and some slave nodes with the help of Amazon EC2. Spark has a good data parallel architecture and point data is well suited for parallel processing. This is the motivation behind using data parallelism here. The OS in each node was Ubuntu 14.04. Scalability was observed to be good in the case of growing point data and number of nodes.

3.2.8 Hybrid Classification. Hybrid techniques in classification are the methods where two or more approaches are combined to produce more efficient application-specific approaches for data mining.

A hybrid approach has been proposed for large-scale data classification over the cloud by Apexa Kamdar et al. [40] based on Naive Bayes and Support Vector Machine-SVM. Classification requires high storage for maintaining intermediate and final results and this requirement can grow exponentially and it can be very difficult to meet the requirements. So, cloud can be an ideal case in which data storage facility can be rented on demand as services for any customized purpose. The algorithm first initializes weights for training examples and creates new dataset using replacement technique. The calculation of prior and conditional probabilities of the dataset takes place. Probabilistic models are used to classify the training example data in such cases. The updating of weights of classified examples takes place as per their classification accuracy. The process of creating data sets is repeated till the training data is classified correctly. As training of data is involved and repeated creation of data sets is done, data parallelism is a great option here. Finally the newer examples are classified using classifier's votes. The output of this stage is fed into SVM for further processing and grouping. This approach has been implemented in Hadoop map-reduce framework successfully providing high accuracy, efficiency, and optimized throughput. Table 11 makes a comparison of various Classification algorithms implemented in Cloud environment discussed in the above section. The table also discusses about volume, velocity and variety aspects of big data for each of the said algorithms in different columns.

Apart from the above categorization, we intend to mention some classification algorithms for streaming data mining that can be implemented over the cloud using the available frameworks and platforms as cited in Section 2.6. Some of the Decision Tree-based approaches are ITI (Incremental Tree Induction) [193], VFDT (Very Fast Decision Tree Learner) [193, 201], Streaming Ensemble Algorithm [193, 202] (VFDT and Streaming Ensemble Algorithm being based on Concept Drift) [193], UFFT (Ultra Fast Formation Tree System) [193], and Random Forest-based Classification Algorithm (using Random Forest) [193, 203]. Nearest Neighbor-based algorithms are Adaptive Nearest Neighbor Classification Algorithm [193] and Any Time Nearest Neighbor Algorithm [193]. Algorithms using Naive Bayes techniques are Weighted Classifier Ensemble [193] and Evolving Naive Bayes [193]. Some other approaches are mentioned in Reference [193].

Table 11. Comparison of Classification Algorithms Implemented in Cloud in Chronological Order

Algorithms and Citations	Year of implementation in Cloud	Cloud Framework used	Optimized for	Advantages	Noise Handling	Parallelization approach	Types of managed data	Scalability
Parallel implementations of KNN, Naive Bayes Classification and Decision Tree based Classification [180] (Citations-59)	2010	Hadoop Map-Reduce	High accuracy in Big data Classification	Good scalability and Parallelism	Excellent	Input Dataset partitioned	Structured data	Linear Scalability
Naive Bayes Classification [36] (Citations-39)	2012	Hadoop with HDFS	High perfection in data Classification	Good latency and Parallelism	Excellent	Data parallelism	Structured data	High (1 million data samples with 11 attributes)
Genetic Algorithm for Cloud [37] (Citations-17)	2012	Map-Reduce with GFS/HDFS	Mitigation of Manual effort in Mining	Maximum Efficiency and Accuracy	Good	Task parallelism in handling subset of data individually	Structured data	Medium (lacs of instances and hundreds of attributes)
PPKC (Privacy Preservation KNN based Classification) [38] (Citations-2)	2012	Cloud General Framework with Cloud Data	Security of Business Data while Mining	Accuracy and Efficiency	Not Known	Data parallelism using local classification	Structured and Semi-structured	Medium
Neural Network based Classification Technique in Cloud [83] (Citations-6)	2012	General cloud setup	Medical data analytics	Visualization and Collaboration in cloud	Good	Not parallelized	Structured medical data	Medium
Class-based micro-classifier ensemble classification technique (MCE) [202] (Citations-23)	2012	Map-reduce	Big data streams with high number of classes	Very good accuracy and speedup	Good	Data stream parallelism	Structured	High
Cloud Support Vector Machine (SVM) [34] (Citations-35)	2013	Cloud Storage and Compute	Huge Volume Dataset training in mining	Excellent scalability and concurrency, high level of parallelism	Good	Data parallelism at training stage	Structured and Un-structured	High (thousands of data instances and around 40 attributes)
SPRINT for Cloud [35] (Citations-2)	2013	Map Reduce Framework	High Parallelism using Multiple Nodes	Scalability and Parallel Implementation	Excellent	Data parallel systems	Structured	High
Hybrid Technique with Naive Bayes and SVM [40] (Citations-3)	2014	Hadoop Map reduce	Good Throughput without unwanted noise	High accuracy, efficiency and optimized throughput	Excellent	Data parallelism	Structured	High
KNN Based Classification [39] (Citations-NA)	2015	Cloud Parallel Model	Privacy of User's information	Security and Efficiency	Not Known	Not parallelized	Structured	Medium
Point Data Classification Technique in Cloud [41] (Citations-7)	2015	Apache Spark	Big Point Data Classification for Geospatial Application	Highly Scalable with addition of nodes to the setup	Good	Data parallelism	Structured	High (3 million data points)
EVFDT [201] (Citations-11)	2015	Cloud environment	Detecting Distributed Denial of Service Attack	Streaming data classification	Good	Data parallelism	Structured	Low
A parallel random forest algorithm for big data in a spark cloud computing environment [203] (Citations-47)	2017	Apache Spark	Data allocation and task scheduling mechanism	Very good accuracy and speedup	Good	Task and data parallelism	Structured	High (100 nodes and up to 2500 GB datasets)

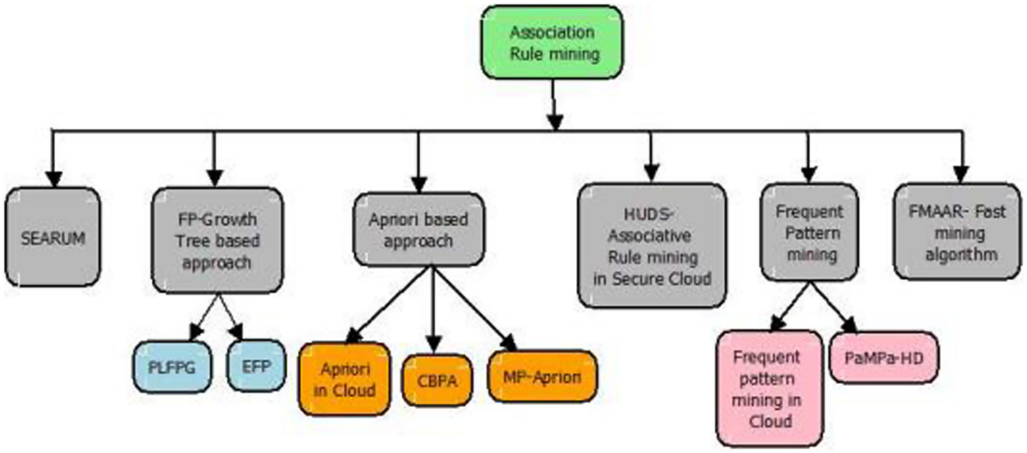

Fig. 21. Taxonomy of association rule mining techniques implemented in cloud.

3.3 Association Rule Mining

Association rule mining or simply rule mining is a technique where relationships are established between different data objects based on some criteria or quantitative threshold also called measure of interestingness. It is widely used in market analysis in business prediction by finding relationships between various items.

Mining rules and learning new patterns can be achieved with a variety of approaches. Some approaches adopted in cloud are FP-Growth tree-based approaches containing PLFPG [48] and EFP [54] algorithms. Apriori-based approach has cloud based Apriori [43], CBPA [47], and MR-Apriori algorithms [42, 179]. Frequent pattern mining techniques have special implementations in cloud [44, 45] and PaMpa-HD [53] is a special parallel implementation. SEARUM [51], HUDS [52], and FMAAR [55] are some other techniques for association rule mining in cloud. Figure 21 shows a classification of various Association Rule Mining techniques implemented in Cloud.

3.3.1 FP-Growth Tree-based Approach. FP-Growth tree is a scalable data structure that helps in mining frequent patterns and item sets with less database scan and candidate generation [104–111]. Lijuan Zhou et al. [48] have proposed a FP-Growth base cloud data mining algorithm for parallel implementation using linked list based on Map reduce (PLFPG). The traditional FG-Growth algorithm calculates the support in datasets and stores the compressed result in database. Later the database is scanned and calculated for item sets. Now, if the data set is huge and a large volume of data is being processed, then the data confidence is calculated and all the value s are stored as big data in database. It is very inefficient and time consuming until parallelized. Hadoop has opened new ventures for traditional FP-growth algorithm by parallelizing the data processing of the entire dataset. The data is fed into many sub nodes from master node and is processed in parallel fulfilling the requirement of faster implementation of FP-Growth algorithm using parallel linked list and enhancing the storage utilization efficiency as compared to normal implementation. It is reported to have better scalability, extensibility, efficiency, and tolerance.

Al-Hamodi et al. [54] published research based on yet another cloud version of FB-Growth algorithm named as Enhanced FP-Growth (EFP) implemented in Hadoop environment as developed in Java. The dataset is divided into almost equal parts with same number of transactions. The EFP-Growth runs over Map-reduce framework. Each of the Mapper function is given a specific subset of data to be processed. The reducer function is assigned with group specific transactions.

The local FP-Growth tree is formed for each subset. The EFP-Growth tree is formed later on the basis of conditions upon the local FP-Growth trees. Hence, the procedure discovers the required patterns. The implementation details can be found in the full paper. The main advantage of this algorithm over FP-Growth tree is its fast execution time in the case of lower threshold.

3.3.2 Apriori-based Approach. Many parallel implementations of Apriori algorithms [90, 91] have been realized in cloud environment using Hadoop [3] and Map-reduce [1, 2] platforms.

Zhang Danping et al. [42] have introduced a cloud version of Apriori algorithm. The Apriori algorithm is famous for finding association rules in diversified data. Parallel version of Apriori (MP Apriori) has been implemented in Map-reduce framework. The frequent item set selection and support and confidence calculation is done in parallel for any given data set with transactions. This helps in reducing the execution time and supports high scalability in the case of huge data volume mining. Hadoop tool has been incorporated to realize this technique for better usage and user experience.

Juan Li et al. [43] have further suggested a new technique of Apriori implementation in cloud map-reduce framework. Analysis of Big Data and mining of rules and establishing relationships in such huge volume of data is a real pain. Such fully unsolved problems are being constantly investigated by scientists over the world daily. The serial implementation of Apriori has been successfully converted into parallel execution using Amazon EC2 tool in Map-reduce framework.

It is always very handy to work on pruned datasets rather than large datasets with lots of unwanted and noisy outliers and conditions. Data pruning is a preprocessing step in an entire data mining task. In case of item set mining in data, it is highly demanding to work upon a data set having more or less the exact required data. Nidhi Khurana et al. [47] have identified an improvement in Apriori algorithm that is capable of pruning candidate data sets using a count-base method and hence the total data scans over the set are reduced. The Cloud Base Pruning Algorithm (CBPA) is an improvement over Apriori algorithm in cloud. The Sector-sphere framework has been utilized for implementing this algorithm in cloud. The sector storage is used in storing the frequent item sets for further calculation. The sector storage is advantageous in the cases of reliability, accessibility, and availability. The sphere compute is used to perform computational services and is working on top of sector storage tier. Some APIs are available for end users to execute parallel application and process data as pre requirement. The entire system works over wide area network. The performance of this algorithm in sector-sphere paradigm is based on the locality of data parameter. If the data is located nearer to the compute node, then the computation will be fast based on the proximity of the datasets and if the data is present in some other sector storage located in foreign jurisdiction, it might take a while in data communication and transportation hence delaying the compute function and result sharing.

3.3.3 Frequent Pattern Mining-based Approach. Kuldeep Mishra et al. have proposed an algorithm for frequent pattern mining in cloud data mining applications [44]. The main concern in mining rules in frequent item sets in datasets is the space required for the frequent item sets. For large datasets, the frequent item sets can be very high and with high amount of frequent item sets it is required that we have huge space to temporarily store them while processing. Space does not come cheap. So, our researcher has rented space from cloud services in low cost to suffice the purpose of generating item sets as required.

Dheresh Soni et al. [45] have applied algorithms in cloud framework for frequent item sets mining. It has the flexibility to operate in non-cloud environment as well. The cost in cloud as well as non-cloud environment has been calculated and efficient use of cloud space services has been made.

Daniele Apiletti et al. [53] further proposed a pattern miner technique that can run in parallel for data with high dimensions named as PaMPa-HD. The algorithm is a parallel implementation of frequent closed item set mining algorithm based on carpenter [167]. The implementation is reported to have load balancing and robustness properties to the best. It has improved the execution time. The carpenter algorithm is famous for mining patterns in datasets with less number of transaction but the transaction have relatively larger number of items per transactions. Carpenter works on depth-first transaction approach. The input dataset is transposed before the enumeration approach is applied. The algorithm is integrated with distributed Map-reduce function to facilitate cloud. The Mapper function processes each row and sends the transposed version to the reduction function. The reduction function executes the locally available carpenter while memory is not exhausted. The second Mapper function mines all the closed item sets or transposed version of the data and sends to the reduction function. Here the data is attained from the previous step. In the second reduce function the eldest of item set in closed frequent set is kept in depth first manner. This is continued until no more data is to be processed. Running time and scalability is improved in distributed execution of this technique and also the curse of dimensionality has been very effectively coped in this algorithm which is a major concern in many FP-Growth-based algorithms.

3.3.4 Other Algorithms. Daniele Apilietti et al. [51] have introduced a novel technique of rule learning in cloud. SEARUM (Cloud-based Service for Association Rule Mining) is a paradigm executed in parallel environment to find interesting relationships and correlation among data. The algorithm is a combination of Map-reduce jobs running in parallel. Each job is a task in the association rule mining process. Each job runs in parallel and gets input from the previous task in the series and provides output to the next task in line.

Mangayarkkarasi et al. [52] have proposed an Associative Rule mining technique in secure cloud environment using rank-based evaluation of datasets with assigned weights as per their frequency of access by users. The model is perfect for predicting frequently accessed data in cloud. Apriori rule mining algorithm used in this case is High Utility Discovery Data Net (HUDS). The management of servers, mining software, storage hardware and computing nodes are centralized. The framework utilized is capable of reducing processing delay by identifying the best resources for the query submitted by users in cloud and also ensures security for data and users with proper authentication services. The cloud data stores are used as services in storing and retrieving meaningful information. After heavy experimentation, it has been observed by the researchers that very relevant information has been successfully mined about customers and their interests. The customer habits and the trend of accessing products as well as needs per customers can also be viewed.

Bo He [55] has produced yet another implementation of cloud mining using association rule mining algorithm called FMAAR—Fast Mining Algorithm of Association Rules. The experimentation on this algorithm is done with respect to normal Apriori algorithm. The results have shown that in the condition of same minimum support, the FMAAR algorithm outperforms the Apriori in the cases of communication traffic and latency. It has also established its superiority in scalability and efficiency while mining long as well as short frequent patterns in huge datasets. The algorithm can be viewed in the full paper.

Web is a warehouse of information and the amount of digital data in web is increasing exponentially. To get a hold of the useful information and the user specific information out of the vast expense of data, a new approach has been proposed by Wenzheng Zhu et al. [56]. The approach is based on cloud computing framework. The Map-reduce framework has been used to implement this approach. A cloud setup has been established having Master Node, Name Node, Algorithm Nodes, and Data Nodes. The name node is the data distribution point that partitions

the data into sub-parts and sends it to data nodes. The algorithm node has the algorithm to be applied. The Master node is the controller of the entire workflow. This technique has been compared with traditional data mining approach and it has been observed that this approach is highly efficient in the case of execution time complexity.

Lingjuan Li et al. [49] have conducted experimentation in rule mining algorithm for cloud. Hadoop is used to examine the algorithmic efficiency in cloud environment. The cloud environment is consisting of a number of nodes and is immured to system failure and tolerance. The number of nodes is varied and the results are recorded to check its impact. With the increase in data sets the number of items in candidate set increases. All these data are difficult to be managed and retrieved in a single node, hence a cluster of nodes is an option. Here the researchers have also provided scope of future enhancement as the number of minimum support values can be set. Parameter sweeping paradigm can be implemented in this regard. User interested and specific association rules can also be facilitated by the algorithm. The execution time is minimized and memory requirements can be optimized as cloud services.

Table 12 makes a comparison of various Associative Rule Mining algorithms implemented in Cloud environment discussed in the above section. It also categorizes the data on the basis of parallelization approaches, types of managed data and scalability features. The main lacking point in Association rule mining algorithms is that much less research has been carried out for real time data mining in this domain. Researchers should focus on the velocity aspect of big data in this domain. Table 13 compares the cloud data mining frameworks in this survey with respect to the performance of data mining tasks. Figure 22 is a categorization of the surveyed algorithms (Clustering, Classification and Association rule mining) on the basis of Parallelism approach (Task, Data, Graph parallelism), Types of Data managed (Structured, Semi-structured, and Un-structured), which signifies the *Variety* aspect in Big data, Scalability (High, Medium, and Low), which portrays the *Volume* aspect of Big data, and Processing support (Batch and Streaming), which is the *Velocity* aspect in Big data.

3.4 Researches in Cloud Data Mining: A Bibliometric Analysis and Grouping

This section is a brief survey on the various major data mining algorithms implemented in cloud in the past 8 years. The current standing of Cloud data mining research and implementation has been placed in the below tables and graphs. According to the survey the research for Cloud data mining algorithms and frameworks gathered momentum in *2008*. Table 14 gives a year-wise segregation of major algorithms implemented in cloud from *2008* to *2017*. It is evident from the below survey that several implementations have been done in cloud environment in the years *2012*, *2013*, *2014*, and *2015*. It can be seen that between 2012 and 2014 many data mining algorithms have been successfully implemented in cloud environment and between 2015 and 2017 to date, the figures are quite satisfactory too. Table 15 clearly shows the utilization of the cloud data mining frameworks surveyed in this article. The applicability of each of the recognized framework has been shown by compiling the numbers of data mining algorithms supported by the framework. This clearly shows that Map-reduce and Apache Hadoop frameworks are mostly used for the purpose. Table 16 shows the current advancements in the three major tasks of data mining: Clustering, Classification and Association Rule Mining, the number of algorithms implemented in all the three categories over the cloud.

4 SECURITY-RELATED RESEARCH ADVANCEMENTS FOR CLOUD DATA MINING

The security aspect is the most vital entity in cloud with the migration of all the organizational data and processes in cloud. Security and privacy of user's data is one of the most researched aspects

Table 12. Comparison of Association Rule Mining Algorithms Implemented in Cloud in Chronological Order

Algorithms and Citations	Year of implementation in cloud	Cloud Framework used	Optimized for	Advantages	Noise Handling	Parallelization approach	Types of managed data	Scalability
Apriori in Cloud [43] (Citations-40)	2012	Map Reduce in Amazon EC2 and Hadoop	Dense Datasets	Scalable and Robust	Good	Data parallelism	Structured	High (about 1 lacs instances and 26 instances)
FMAAR- Fast Mining Algorithm of Association Rules [55] (Citations-1)	2012	Cloud Storage and Compute Environment	Long as well as Short frequent patterns in Huge Datasets	Efficiency and Scalability	Good	Not parallelized	Structured	Medium
Frequent Pattern Mining in Cloud [44] (Citations-9)	2013	Cloud Storage and Compute System	Large amount of frequent patterns	Parallelism and Cloud Storage Efficiency	Good	Not parallelized	Structured	Low
Improved Apriori Algorithm (CBPA) [47] (Citations-NA)	2013	Sector-Sphere Framework	Pruning Task in Mining	Reliability, Accessibility, Availability and Data locality	Excellent	Data parallelized	Structured	Medium
SEARUM [51] (Citations-22)	2013	Task Parallel Implementation of map Reduce	Horizontal Scalability	Optimum Throughput and Latency	Good	Task/Job Parallelism	Structured data	Highly scalable
PLFPG [48] (Citations-14)	2014	Hadoop with HDFS	Storage utilization efficiency and Computing Economy	Scalability, extensibility, efficiency and tolerance	Good	Data Parallel approach	Structured data	Highly scalable (1 lac to 20 lacs data instances)
Cloud Based Apriori MR-Apriori [42] (Citations-7)	2014	Map reduce	Highly Diversified Data	High Scalability and User Friendly Interface	Good	Data parallelism	Structured data	High (10 GB to 30 GB, 100 million transactions)
Cloud Based Apriori MR-Apriori [179] (Citations-31)	2014	Map reduce Hadoop	Highly Diversified Data	High Scalability and efficiency	Good	Data parallelism and distribution	Structured data	High (around 1 lacs data instances, scales very well with nodes increasing from 5 to 25)
PaMPa-HD [53] (Citations-5)	2015	Map reduce Framework	High-dimensional Data Classification	Load Balancing, Robustness	Good	Task parallelism in DFS of sub-trees	Structured and semi-structured data	High (1 million to 10 million items and 5 lacs to 50 lacs items per transactions)
EFP [54] (Citations-2)	2016	Hadoop with Java	Discovery of patterns accurately	Parallel Implementation and Accuracy	Good	Data parallelism	Structured data	High (around 10 lacs transactions with 41 thousand items approximately)
Frequent Pattern Mining in Cloud [45] (Citations-NA)	2016	Cloud Storage and Compute Environment	Flexibility of use with any environment	Efficient use of Cloud Storage	Good	Not parallelized	Structured	Low
Associative Rule mining technique in secure cloud HUDS [52] (Citations-NA)	2017	Cloud Data and Parallel Compute Framework	Frequently accessed Data in cloud	Customer Oriented Services, Accurate Classification Results	Excellent	Not parallelized	Structured data	Medium (data ranging from few thousands to a lac)

Table 13. Performance of Cloud Data Mining Framework Based on Data Mining Tasks

Cloud Data Mining Frameworks	Clustering	Classification	Association Rule mining
Map reduce (2004) [1, 2]	Excellent	Excellent	Excellent
Distributed Graph lab (2012) [11, 12]	Good	Bad	Bad
Parameter Sweeping (2011) [10]	Good	Bad	Good
Sector Sphere (2008) [6]	Good	Bad	Good
Spark (2009–10) [9]	Excellent	Good	Bad
Hadoop (2005) [3]	Excellent	Excellent	Excellent
Pegasus (2009) [95]	Good	Good	Good
Airavat (2010) [59]	Good	Good	Bad
FSBD (2014) [94]	Good	Good	Bad
Pregel (2010) [96]	Good	Bad	Bad
Giraph (2015–16) [97–99]	Good	Bad	Bad
GraphX (2013–14) [100–102]	Good	Good	Bad
Mahout (2017) [130, 131]	Excellent	Bad	Bad

in cloud computing environment. Researchers have come up with some facilities for maintaining data privacy [125].

Cloud data mining and security can be divided into the following aspects: Users Data Privacy, Users Identity, Encryption, and Policy Compliance [84]. Data privacy is the primary motive of secure cloud environment. Data from one organization must not be viewed by people from another organization. Users performing various data mining tasks in the cloud environment must be identified and verified by the system before granting access. Unauthorized users must not be allowed into the vast expense of digital data of an organization. Proper encryption of private and confidential data must be ensured for restricting unwanted users from viewing the same. Policy compliance is a major concern in the cloud data scenario. Policies may vary from organization to organization and to comply with those policies in required level is a challenge.

Airavat [already discussed in Section 2.1.1] is a model atop Map-reduce framework for data mining tasks that is capable of maintaining strong security as well as privacy of user's data [59]. Airavat is capable of performing user specific computations of data and maintain the required level of data abstraction protecting its identity as per policy. A differential privacy [60–63] mechanism is made to use for protecting data and information from unauthorized users. All the data are tagged with appropriate access control stamps. The output will hold the union of all the data tags. K-means and Naive Bayes are implemented successfully over this architecture. Jian Wang [38] and Pooja Bajare et al. [39] have proposed privacy preservation mining algorithms on classification as already discussed in Section 3.2. Mangayarkkarasi et al. [52] have proposed an Associative Rule mining technique in secure cloud environment that has been elaborated in Section 3.3. Hanna M. Saidet et al. [64] researched about the security aspect of cloud mining system, Risks and threats that prevail in a cloud-based processing system. The researcher has used various paradigms to recognize attacks and threats on cloud data using Naive Bayes, SVM, Decision Tree and Neural Networks.

A way to control the privacy and security in Cloud data processing is distribution. Cloud distributors and Cloud providers are the key. Client data can be given to distributors and they can divide the data into partitions. These partitions can be provided to service providers. This way the

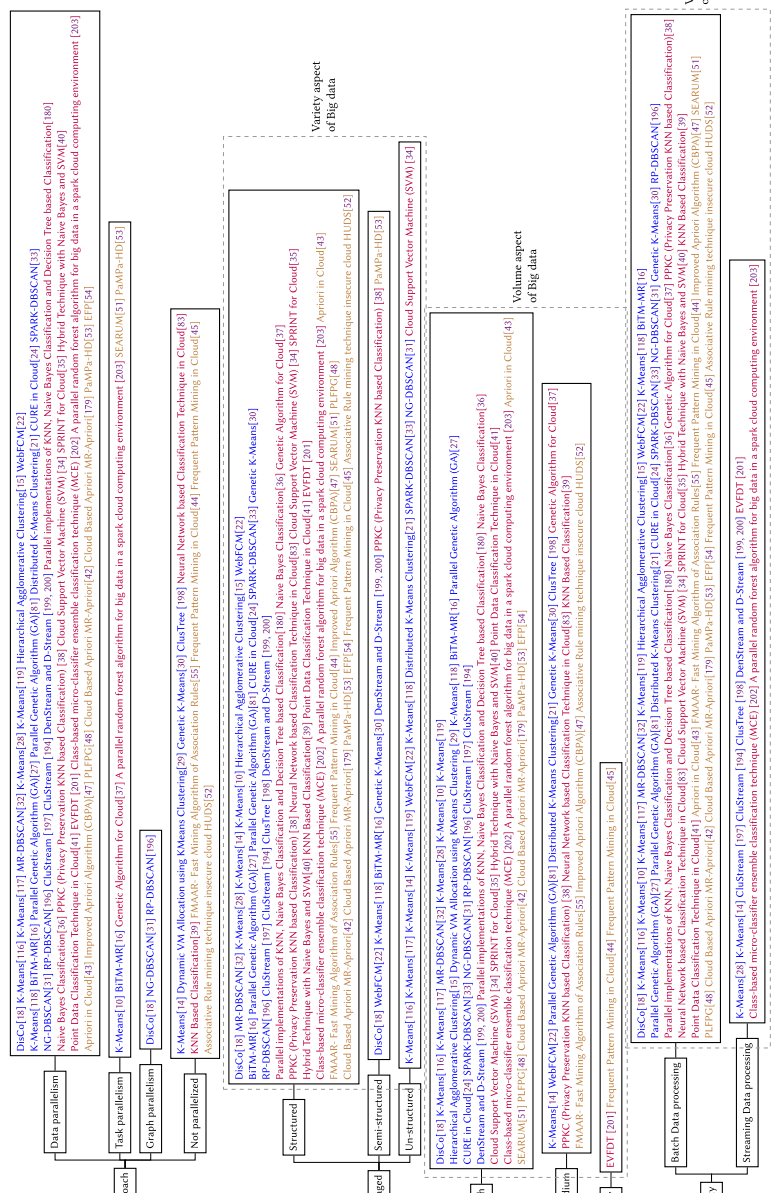

Fig. 22. The categorization of all the surveyed clustering (in blue), classification (in purple) and association rule mining (in brown) algorithms on the basis of parallelism, types of data managed, scalability, and batch/stream processing support.

Table 14. Year Wise Compilation of Data Mining Algorithms Implemented in Cloud

Year of Proposal and Implementation	Algorithms and Techniques	Total
2008	DisCo [18], Hierarchical Clustering in Cloud [25], Parallel Bi-Clustering [17]	3
2009	K-Means [116]	1
2010	Parallel K-NN, Naive Bayesian Model and Decision Tree-based Classification [180]	1
2011	MR-DBSCAN [32], K-Means Clustering [10, 28, 117], Association rule mining in cloud [49]	5
2012	K-Means [14, 119], FMAAR- Fast Mining Algorithm of Association Rules [55], Apriori in Cloud [43], PPKC (Privacy Preservation KNN based Classification [38], Genetic Algorithm for Cloud [37], Naive Bayes Classification [36], Neural Network based Classification [83], Class-based micro-classifier ensemble classification technique (MCE) [202]	9
2013	Hierarchical Agglomerative Clustering [15], Web FCM [22], Dynamic VM Allocation using KMeans Clustering [29], Cloud SVM [34], SPRINT for Cloud [35], Frequent Pattern Mining in Cloud [44], Improved Apriori Algorithm (CBPA) [47], SEARUM [51], MR.SCAN [139], Parallel K-Means [26], CluStream [194]	11
2014	PLFPG [48], Cloud-based Apriori MR-Apriori [42, 179], Hybrid Technique with Naive Bayes and SVM [40], Bi-Clustering BiTM-MR [16], K-Means [118], Hierarchical Clustering in Cloud [19], Association rule mining in cloud [56]	8
2015	Distributed K-Means Clustering [21], CURE in Cloud [24], Parallel GA [27, 81], KNN-based Classification [39], Point Data Classification Technique in Cloud [41], Rule Mining Algorithm for Cloud [43], PaMPa-HD [53], Den-Stream and D-Stream [199, 200], EVFDT [201]	11
2016	EFP [54], Genetic K-Means [30], NG-DBSCAN [31], SPARK- DBSCAN [33], K-Means [30], Parallel implementation of Apriori [90, 91], Frequent Pattern Mining in Cloud [45], CluStream [197]	9
2017	Associative Rule mining technique in secure cloud HUDS [52], A parallel random forest algorithm for big data in a spark cloud computing environment [203]	2
2018	RP-DBSCAN [196], ClusTree [198]	2

data can be distributed across many providers and hackers cannot get hold of the entire data to derive any meaningful information out of it [85, 86].

Cloud security for data mining tasks can be achieved as a service like the other services in cloud (SaaS, PaaS, and IaaS). DSaaS-Data Security as a Service [87] can be availed as a cloud service for ensuring security and privacy of data in cloud environment. Key management and access controls are managed as cloud services. The Cloud security service providers need to be trusted on such scenarios. Again, there are issues if the service providers are dishonest.

5 PARALLELISM, DISTRIBUTION, APPROXIMATION, AND QUANTUM COMPUTE IN CLOUD DATA MINING

Parallelism and distribution are the two important techniques that are utilized to make Cloud data mining more demanding and efficient than classical data mining. Parallelism denotes the process of parallel execution of a task on multiple partitions of a dataset or parallel execution of multiple

Table 15. Framework Wise Segregation of Data Mining Algorithms in Cloud

Framework for Cloud Data Mining	Algorithms and Techniques	Total
Map reduce (2004) [1, 2]	K-Means [14, 116–119], MR-DBSCAN [32], CURE in Cloud [24], Parallel GA [27, 81], Genetic K-Means [30], SPRINT for Cloud [35], Genetic Algorithm for Cloud [37], Cloud-based Apriori MR-Apriori [42, 179], Apriori in Cloud [43], SEARUM [51], PaMPa-HD [53], MR.SCAN [139], Parellel implementation of Apriori [91], Association rule mining in cloud [56], Parallel K-Means [26], Class-based micro-classifier ensemble classification technique (MCE) [202]	22
Parameter Sweeping (2011) [10]	Rule Mining Algorithm for Cloud [43, 52], K-Means [10]	3
Sector Sphere (2008) [6]	Genetic K-Means [30], Improved Apriori Algorithm (CBPA) [47]	2
Distributed Graph lab (2012) [11, 12]	NG-DBSCAN [31]	1
Spark (2009-10) [9]	SPARK- DBSCAN [33], Bi-Clustering BiTM-MR [16], Point Data Classification Technique in Cloud [41], RP-DBSCAN [196], CluStream [197], A parallel random forest algorithm for big data in a spark cloud computing environment [203]	6
Hadoop (2005) [3]	K-Means [14, 116–119], DisCo [18], Distributed K-Means Clustering [21], Parallel GA [27, 81], Naive Bayes Classification [36], Hybrid Technique with Naive Bayes and SVM [40], Apriori in Cloud [43], PLFPG [48], EFP [54], Parallel implementation of Apriori [90], Association rule mining in cloud [49, 50], Parallel K-NN, Naive Bayesian Model and Decision Tree-based Classification [180]	18
Generalized Cloud Compute Setup	Hierarchical Agglomerative Clustering [15], K-Means Clustering [28], Web FCM [22], Dynamic VM Allocation using KMeans [29], Cloud SVM [34], PPKC (Privacy Preservation KNN-based Classification) [38], KNN-based Classification [39], Frequent Pattern Mining in Cloud [44, 45], Associative Rule mining technique in secure cloud HUDS [52], FMAAR—Fast Mining Algorithm of Association Rules [55], Hierarchical Clustering in Cloud [19], Agglomerative Clustering in Cloud [20], Hierarchical Clustering in Cloud [25], Parallel Bi-Clustering [17], Neural Network-based Classification [83], CluStream [194], ClusTree [198], DenStream and D-Stream [199, 200], EVFDT [201]	21

tasks of an algorithm. Distribution is the process of dividing the tasks into smaller tasks and allows *N* number of nodes to execute these tasks at the same time. Both of these techniques are prevalent in Cloud data mining [88, 164].

5.1 Task or Data Parallelism?

Parallelism in mining tasks can be achieved in two ways: data parallel execution and task parallel execution. In data parallel execution, it's quite obvious that the mining algorithm will execute independently upon the partitions of the originally huge dataset in different compute sites. The local results will be merged to form the global output. In the case of task parallelism, either the algorithm will run in many streams independently with different parameters and attributes or the algorithm will be divided into sub procedures, and the procedures will run parallel at the same time in different sites. The second one is acceptable only if there is no dependency of execution between the sub-procedures of the algorithm. Many frameworks are implemented to support

Table 16. Task Wise Segregation of Data Mining Algorithms in Cloud

Data Mining Tasks in Cloud	Algorithms and Techniques	Total
Clustering [Section 3.1]	DisCo [18], MR-DBSCAN [32], MR.SCAN [139], K-Means Clustering [10, 14, 28, 116–119], Hierarchical Agglomerative Clustering [15], Hierarchical Clustering in Cloud [19], Agglomerative Clustering in Cloud [20], Web FCM [22], Dynamic VM Allocation using Kmean [29], Bi-Clustering BiTM-MR [16], Distributed K-Means Clustering [21], CURE in Cloud [24], Parallel GA [27, 81], SPARK-DBSCAN [33], NG-DBSCAN [31], Genetic K-Means [30], Hierarchical Clustering in Cloud [25], Parallel K-Means [26], Parallel Bi-Clustering [17], Frequent Pattern Mining in Cloud [45], CluStream [197], CluStream [194], ClusTree [198], DenStream and D-Stream [199, 200]	32
Classification [Section 3.2]	Naive Bayes Classification [36], Genetic Algorithm for Cloud [37], PPKC (Privacy Preservation KNN-based Classification) [38], Cloud SVM [34], SPRINT for Cloud [35], Hybrid technique with Naive Bayes and SVM [40], KNN-based Classification [39], Point Data Classification technique in Cloud [41], Neural Network-based Classification [83], Parallel K-NN, Naive Bayesian Model and Decision Tree-based Classification [180], EVFDT [201], Class-based micro-classifier ensemble classification technique (MCE) [202], A parallel random forest algorithm for big data in a spark cloud computing environment [203]	13
Association Rule Mining [Section 3.3]	Frequent Pattern mining in Cloud [44, 45], Apriori in Cloud [43], FMAAR-Fast Mining algorithm of Association rules [55], Improved Apriori Algorithm (CBPA) [47], SEARUM [51], PLFPG [48], Cloud-based Apriori MR-Apriori [42, 179], PaMPa-HD [53], EFP [54], Association Rule mining technique in secure cloud HUDS [52], Parallel implementation of Apriori [90, 91], Association rule mining in cloud [49, 50, 56]	17

data parallelism. Map Reduce and Distributed Graph Lab are such frameworks. Parameter sweeping [10] and Sector Sphere [6, 7, 69] are capable of implementing task parallel mining jobs. More frameworks needs to be researched for data parallel implementations for Big Data mining in cloud maintaining accuracy and consistency for higher dimensional diversified dense data. Task parallel frameworks need to be improved to gain better insight about the mining results and fault tolerance in cloud. The researchers can also find new ventures in merging the concept of data as well as task parallelism and creating hybrid frameworks for better efficiency and attaining better time complexity. However, this will demand better storage and compute facilities over the cloud, and maintaining consistency over high distribution is also a concern. Security of user data is also a major topic of investigation in cloud mining. Much needs to be done from cloud as well as mining end to overcome this situation of privacy and security of data and information over the cloud.

5.2 Vertical vs. Horizontal Scalability?

Vertical scalability refers to the process of adding resources such as CPU and RAM to a single computing node for achieving high degree of task/data parallelism. However, horizontal scalability is the technique of achieving the parallelism over the network with more than one computing node operating on the data as a single computing recourse. Vertical scalability lies in the realm of shared memory whereas Horizontal scalability attributes to local memory per node. Both has its advantages and disadvantages. Vertical scaling (homogeneous) is ideal if the amount of parallelism and scalability is limited but it requires downtime of organizational data-centers. In case of mining tasks carried out by data centers, vertical scaling was preferred before the advent of cloud

computing. Now, that cloud data mining has gained importance due to its flexibility and speedup, vertical scalability is not preferred over horizontal scalability (homogeneous or heterogeneous in nature). However, vertical scalability has advantages such as data locality and minimum communication delay for data and parameter transfer. However, horizontal scalability is very versatile; commodity hardware can be added and removed on demand. But in the case of implementing mining frameworks proper data-practitioner module and load balancer modules must be in place to achieve better results. As the number of computing nodes increases, the problem of data locality increases and communication cost and network latency also increase. So, the new researches can be headed towards rigorous experimentation on optimum setup combining vertical and horizontal scalability in terms of big data mining tasks. Hybrid frameworks might be considered to be proposed and implemented on storage and compute nodes on such setups.

5.3 Intra Node vs. Inter Node Parallelism?

The terms inter node and intra node parallelism refer to the domain of parallelism in cloud mining scenario. A system with multi-core capacity is capable of running algorithms in parallel without multiple compute nodes involved. The amount of data transfer and network overhead can be avoided in such cases contrary to inter node parallelism where data and instructions need to be transported between nodes over network. Fault tolerance can be mitigated in the multi-core intra-node parallelism and task performance can be tuned by minimizing network delays. But limited scalability can be achieved with such an approach. Both have merits and demerits. The best of both approaches can be taken into account by determining the extent of multi-core scalability along with multi-node parallelism to achieve an optimized setup over the cloud for mining tasks [57, 58].

5.4 Exact vs. Approximate Computing?

The concept of approximate computing has been researched for two decades now. Exact computing is the general way a computer processes data accurately without loss of information. Such processing is necessary when the accuracy is a concern and cannot be traded off with latency. With respect to the review, data mining in cloud not only requires newer frameworks that will enhance the existing ones but also needs the existing algorithms to be efficient in energy and time perspective. The current digital data volume may be handled the way it is meant to be but in very near future the nuclear explosion of digital data (contributed by IOTs, Sensors, Medical data, etc.) will be so tremendous that if we rely on exact computation, then we may face severe issues. This has been rightly stated by R Nair in his technical report [133]. Approximate computing [144] methods in big data mining have been adopted recently [134–138] for trading off result accuracy with energy and latency. Not only should the researchers focus on developing and designing newer algorithms to support approximate computing but also try to develop new frameworks (such as ApproxHadoop [165] and IncApprox [188]) for cloud data mining where approximation should be an inbuilt property. Cloud service providers can also think about providing approximate computing as services (*AxCaaS*) to clients as and when demanded for various big data mining tasks.

5.5 Classical vs. Quantum Computing?

Quantum computing [145, 150, 159, 160] is an old paradigm in computer science research. It has gained recent popularity in many fields of computation such as pattern analysis, data mining and artificial intelligence [146]. Classical computers work on the basis of two transistor states “0” or “1” called bits, and the transistor can be in any one of these sates at a time. But in the case of quantum computers, we have superposition of such states and such states are often called qubits. Quantum computing has the advantage of solving some computation problems much faster than their classical counterpart [158]. Many attempts have been made to apply quantum computing techniques to data mining and big data analytics to gain better performance [147–149, 151–153]. Google and

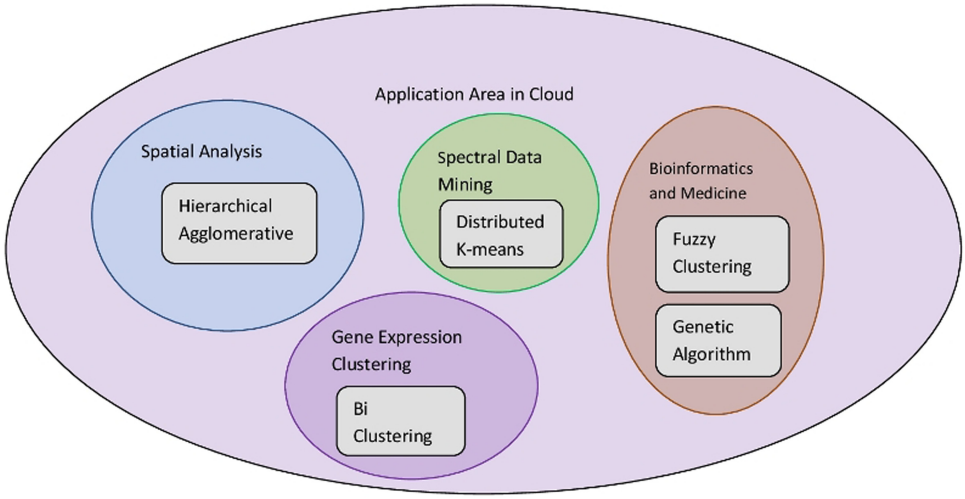

Fig. 23. Grouping of data mining techniques in cloud as per application area.

IBM are actively working on quantum computing research [154–157]. Promising advancements have been made by Google to introduce quantum computing services in cloud. Looking into this, researchers can think of designing frameworks for cloud data mining powered by quantum computing for performance gain. *QCaaS*-Quantum Computing as a Service in cloud can be a promising future for cloud as well as Big data mining opportunities in cloud.

6 APPLICATIONS OF CLOUD DATA MINING

Cloud data mining has been the most researched area in cloud computing over the past 7 years. With the passage of time researchers have come up with many implementations of classical data mining algorithms in cloud environment. These algorithms have been realized in cloud environment with the help of some cloud mining frameworks and paradigms already discussed in the previous sections. This article has identified a variety of application areas for cloud data mining reviewing the papers and technical reports of various algorithmic implementations. We have classified the algorithms on the basis of application areas that are highly important in the perspective of Big data analytics, Business intelligence, Artificial intelligence, Image-processing, Machine-learning Computer vision applications, Anomaly detection, Intrusion detection, Bioinformatics, Geoinformatics, and some more. Figure 23 shows a connection between the cloud data mining frameworks and the application areas where these frameworks have mostly participated. It also shows the algorithms that are successfully implemented by these frameworks for the said applications. Table 17 summarizes the algorithms implemented in the major application areas in cloud and the frameworks utilized for these implementations.

Spatial Data Mining and Geo-Spatial Data Analysis and Geoinformatics: Analysis of space (2D and 3D) data has come up as a big technological requirement for applications like image-processing, pattern analysis, GIS (Geographic Information System), and Google glass. Geo-Spatial analysis and informatics in geology are well know field of data mining dealing with satellite imagery and image camouflage. Geological data analysis has risen to great importance with respect to geographic positioning systems and road detection and extraction techniques. Some of the algorithms implemented in cloud for such application are K-Means [10, 14, 28]; Spark DB-SCAN [33], NG-DBSCAN [31], and RP-DBSCAN [196]. Basically, Cluster analysis is being employed for

Table 17. Cloud Data Mining Algorithms and Their Major Application Areas

Application area	Algorithms and Techniques	Frameworks
Spatial Data Analysis and Satellite Imagery	K-Means [10, 14, 28, 116–119], MR-DBSCAN [32], SPARK-DBSCAN [33], Hierarchical Clustering in Cloud [25], RP-DBSCAN [196]	Map Reduce [1, 2] and Spark [9]
Spectral Data Analysis	Hierarchical Agglomerative Clustering [15], Parallel K-Means [26]	Distributed Cloud Compute Environment, Map Reduce [1, 2]
Gene Expression Data Mining and Bioinformatics	Bi-Clustering, BiTM-MR [16], Parallel Bi-Clustering [17], Neural Network-based Classification [83], A parallel random forest algorithm for big data in a spark cloud computing environment [203]	Map-reduce [1, 2]/Spark Framework [9]
Geo-Spatial Analysis and Geoinformatics	Point Data Classification Technique in Cloud [41], RP-DBSCAN [196]	Apache Spark [1, 2, 9]
2D-Dataset Analysis/nD-dataset analysis	NG-DBSCAN [31], RP-DBSCAN [196]	Distributed Graph-Lab [11, 12]
Large-scale Mining in huge Big Data and Web mining	DisCo [18], Distributed K-Means Clustering [21], Dynamic VM Allocation using KMeans Clustering [29], Genetic K-Means [30], Cloud SVM [34], SPRINT for Cloud [35], Frequent Pattern Mining in Cloud [44], PLFPG [48], Rule Mining Algorithm for Cloud [43], SEARUM [51], Associative Rule mining technique in secure cloud HUDS [52], Association rule mining in cloud [49, 50, 56], Parallel K-NN, Naive Bayesian Model and Decision Tree-based Classification [180]	Hadoop Map-reduce [1–3] and Sector Sphere/Parameter Sweeping [6, 10], A parallel random forest algorithm for big data in a spark cloud computing environment [203]
Statistical Data Analysis	Web FCM [22], Bi-Clustering BiTM-MR [16]	Spark [9]
Data pruning and removing noisy data	Improved Apriori Algorithm (CBPA) [47], CURE in Cloud [23, 24]	Sector Sphere/Map Reduce [1, 2, 6]
Machine-learning Applications	Parallel GA [27, 81], Cloud SVM [34], Naive Bayes Classification [36], Genetic Algorithm for Cloud [37], Hybrid Technique with Naive Bayes and SVM [40]	Hadoop Map Reduce [1–3]
High-dimensional Data Mining	PaMPa-HD [53]	Map Reduce [1, 2]
Highly Diversified and Dense Data Mining in rule mining	Cloud-based Apriori MR-Apriori [42, 179], Apriori in Cloud [43], EFP [54], FMAAR- Fast Mining Algorithm of Association Rules [55], A parallel random forest algorithm for big data in a spark cloud computing environment [203]	Map-reduce Hadoop [1–3]
Security of data in cloud, Anomaly and intrusion detection	KNN-based Classification [38], CluStream [197], CluStream [194], ClusTree [198], DenStream and D-Stream [199, 200], EVFDT [201], Class-based micro-classifier ensemble classification technique (MCE) [202]	Distribute Cloud Compute Environment, Spark [9]

spatial data mining and analysis purpose. Map-reduce [1, 2] and Apache Spark [9] frameworks are known to have implemented such data analysis algorithms. Point data classification techniques [41] have been implemented for Geoinformatics in cloud environment.

Spectral Data Analysis and Gene Expression Data Mining for Bioinformatics: Analysis of spectral data is a nontrivial task in microbiology and bioinformatics. Complex biological systems such as genes and micro-molecular structures are tough to be analyzed and need special algorithmic paradigms. Also the data sets relating such data are huge in size and cannot be efficiently analyzed without parallelism and distribution. Hierarchical-Agglomerative Clustering [15] and Bi-Clustering techniques BiTM-MR [16] are algorithms implemented over cloud frameworks such as Map Reduce [1, 2] and Spark [9]. Generally, Cluster analysis is employed for such applications in cloud.

Higher-dimensional Data Analysis: Higher dimensionality is a curse in data analysis. As the dimensionality of data increases, the process to analyze and classify the data becomes tougher. Basically, medical data, such as blood analysis data, genes data, and human immune system data, are considered very high-dimensional data. In pattern and image recognition or computer vision in robotics, scientists deal with high dimensionality. NG-DBSCAN [31] and PaMPa-HD [53] are some algorithms implemented in cloud using Distributed Graph-Lab [11, 12] and Map-reduce [1, 2] framework, respectively.

Large-scale Mining of Big Data in Cloud: Mining of data is a trivial task if the size of the dataset is not so large. But if the dataset is expected to have terabytes of data, it is indeed a complex procedure. Due to this very notion, researchers have tried to implement data mining in cloud environment to achieve parallelism and distribution over the cloud. Many algorithms in Clustering, Classification and Associative rule mining are implemented over the cloud in this regard. In clustering, DisCo [18], Distributed K-means [21], Dynamic VM Allocation using KMeans clustering [29], and Genetic K-means [30]. In classification, we have Cloud SVM [34] and SPRINT for Cloud [35] implemented for cloud. Associative rule mining techniques implemented for cloud large-scale data mining are Frequent pattern mining in cloud [45], PLFPG [48], Rule Mining algorithms in cloud [49, 50, 56], SEARUM [51], Associative rule mining technique in secure cloud HUDS [52]. Frameworks basically use in this application are Apache Hadoop [3], Map Reduce [1, 2], Sector-sphere [6], and Parameter Sweeping [10].

Machine-learning Applications: Machine learning in Artificial Intelligence is an application of Computer Science, where computer machineries are programmed to learn themselves with respect to some environmental parameters and input data. These paradigms are self evolving when exposed to variety of data. Data mining is an important aspect of machine learning. Supervised and Un-supervised data mining tasks form the backbone of feature selection and extraction in machine learning. Some cloud implemented algorithms in this respect are Parallel GA [27, 81], Cloud SVM [34], Naive Bayes Classification [36], and Genetic Algorithms for cloud [37] and Hybrid techniques with Naive Bayes and SVM [40]. These have been mostly implemented in Hadoop [3] and Map-reduce [1, 2] frameworks. Generally Classification techniques are used in these applications.

Highly Diversified and Dense Data Mining in Rule Mining: Big data is often characterized by the three V's; Volume, Velocity, and Variety, sometimes Veracity is included in this definition. This implies that Cloud data mining often involves processing of Big data, which is voluminous and varied. Such data needs to be handled efficiently or it might lead to unwanted and wrong results. Some algorithms implemented in cloud for handling such data sets are Cloud-based Apriori MR-Apriori [42, 179], Apriori in cloud [43], EFP [54], and FMAAR-Fast Mining algorithm of Association rules [55]. It is to be noted that majority of algorithms implemented for mining diversified data is

based on Association rule mining technique. Hadoop [3] and Map-reduce [1, 2] frameworks have been successfully used for such implementations.

Statistical Data Analysis and Business Intelligence (BI): Data analysis in industries is a vital process as in deriving samples and trends in historic data and bringing out newer insights about the progress of the business through the ages. It is capable of finding optimum solutions and techniques out of data, which can strengthen the business. Some clustering algorithms for statistical data analysis with varied parameters are Web FCM [22] and Bi-Clustering BiTM-MR [16] implemented in Apache Spark [9] in cloud.

Data Pruning and Removing Noisy Data in Cloud: Cloud data bases and warehouses are full of unwanted data and noisy data. It is the responsibility of the client and other users to extract the important data out of the entire datasets to perform the required analysis. Such tasks can be classified as pre data mining tasks that forms the part of actual data mining. Two algorithms surveyed in this article are Improved Apriori Algorithms (CBPA) [47] and CURE in cloud [24] implemented in Sector-Sphere [6] and Map-reduce [1, 2] frameworks. Figure 24 is a complete overview of relationship between CDM frameworks, algorithms, and application areas. It shows a complete mapping of the said attributes in cloud environments to get a whole and comprehensive picture of the scenario.

7 CONCLUSION

The above discussed literature and survey on Cloud Mining Frameworks, Algorithms, and Paradigms convey a few points. It is evident that much work has been proposed in the field since 2004 and this research has gained momentum from 2008. The first framework for mining in cloud is Map Reduce and is the most popular for its distributed approach. Most of the algorithms surveyed in this literature are implemented atop of this framework. But this framework faces serious customer oriented service issues and is not suitable for many mining tasks such as analyzing high-dimensional data and machine-learning tasks using mining. Hadoop is an extension over Map Reduce and it supports application development and enhances parallelism over cloud. It is also reviewed as having supported many mining algorithms such as machine-learning tasks on highly diversified and dense data. Spark is a promising framework that has merged the advantages of Map reduce and the solutions to its problems. It has outperformed Map Reduce in terms of spatial, geo-spatial and statistical mining as well as machine-learning tasks producing better throughput and excellent latency. Distributed Graph Lab further improves the utility of Map Reduce by diversifying the types of algorithmic tasks it can support on graph parallel notion. Many spatial, geo-spatial and high-dimensional data can be mined upon this framework in cloud. Basically these frameworks are capable of implementing mining algorithms in parallelism using distributed computation over data sets. Some applications demand high degree of task parallelism to attain efficiency such as analyzing data with different parameters and parametric values. Parameter Sweeping framework is capable of such parallelism in cloud environment. Large-scale mining is widely carried out in this framework. More algorithms need to be implemented in this framework where higher degree of task parallelism is needed to fulfill analysis tasks with varied parameters. Also from research point of view, computer scientists and researchers can use this framework in designing newer algorithms and experiment with various parameters/values to define their algorithmic limits and parameter considerations. Sector-Sphere framework has been reviewed as a model for noise handling and data pruning activities. Its unique storage and compute facility, in the form of sector and sphere, makes it capable of implementing user defined mining tasks on huge datasets. Security of user's data has been taken care of by many algorithmic implementations. Newer dimension in security of cloud Data Security as a Service (DSaaS) has been introduced to

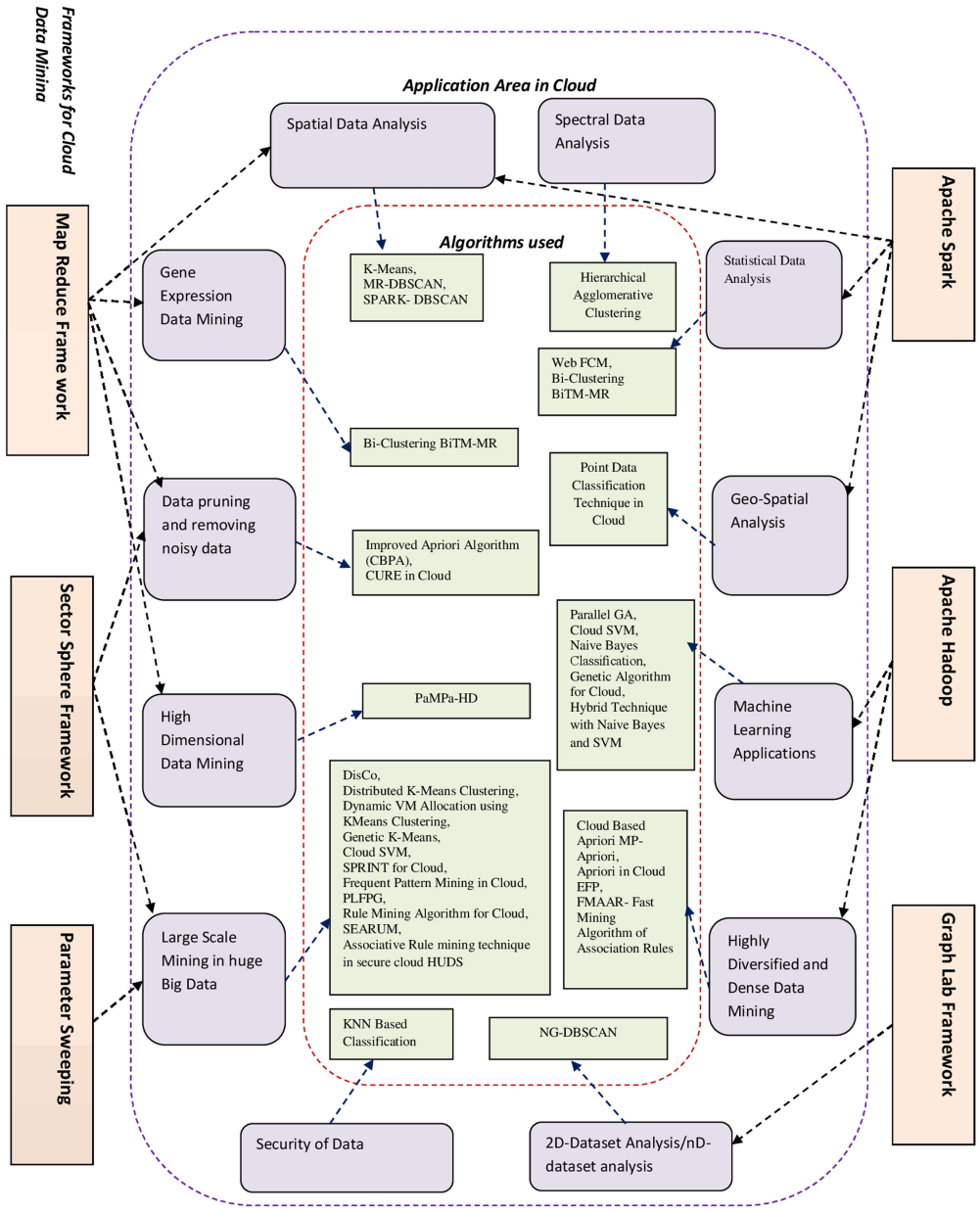

Fig. 24. Relationship between frameworks and algorithms for cloud data mining.

maintain security while performing cloud data mining. Performance improvement initiatives have been taken for some frameworks, such as Map Reduce and Hadoop. But, these implementations can hamper high distribution and scalability in cloud and such optimizations are framework specific and are non-compatible with other frameworks. Optimization techniques must have high degree of compatibility and scalability and future research must be directed towards this area. More focus needs to be given to other frameworks such as Distributed Graph-Lab and

Parameter Sweeping and Sector-Sphere in future to facilitate better data mining tasks. A wide area of applications have been covered by cloud data mining techniques in the past 8 years. Focus has been given on spatial and spectral data mining problems and high diversified and dense data. Big data and large-scale mining have been amply taken care of as per current technology. Much focus still needs to be given in the field of Bioinformatics, Geoinformnatics and Security of data in cloud. Data and task parallelisms have been the key to cloud data mining. Researchers may seek newer approaches in cloud combining the power of data and task parallelism effectively to ensure optimum utilization of cloud services in near future. Apart from the discussed points, we need to think about bringing approximation in big data mining and analytics in cloud. The already implemented algorithms in cloud need to be reviewed and code level changes can be incorporated to approximate the results to enable better performance and energy efficient computing. Approximate Computing as a Service (*AxCaaS*) can be thought of as a whole new avenue for researches to delve and produce better cloud data mining paradigms and frameworks. Apart from this, quantum computing in Cloud data mining can be an innovation that can open up new line of researches in performance and high speed computations in cloud data analytics. Quantum Computing as a Service (*QCaaS*) might lead to newer heights in Big data analytics using cloud. It is also seen that the Volume and Variety aspects of Big data mining (in cloud) have been amply attended by researchers by using scalable and robust algorithms that can handle variety of structured and un-structured data. The Velocity aspect is taken care by streaming data mining techniques but less techniques and methods are proposed in this respect. Researchers can take up this aspect for future research of streaming association rule mining techniques in cloud environments.

ACKNOWLEDGMENTS

The authors thank the editors and anonymous reviewers of ACM CSUR for their valuable feedback and suggestions, which helped in enhancing the quality of the article. They are also grateful to Dr. Arijit Mukherjee and Dr. Himadri Sekhar Paul of TCS Research and Innovation Lab, Kolkata, for their help and support throughout.

REFERENCES

[1] Ronald C. Taylor. 2010. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC Bioinform.* 11, 12 (2010), S1.

[2] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: Simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.

[3] X. Geng and Z. Yang. 2013. Data mining in cloud computing. In *Proceedings of the International Conference on Information Science and Computer Applications (ISCA'13).* 1–7.

[4] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica. 2008. Improving MapReduce performance in heterogeneous environments. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation.* 7.

[5] A. X. Tan, V. L. Liu, M. Kantarcioglu, and B. Thuraisingham. 2010. A comparison of approaches for large-scale data mining. Technical Report UTDCS-24-10.

[6] Yunhong Gu and Robert L. Grossman. 2009. Sector and sphere: The design and implementation of a high-performance data cloud. *Philos. Trans. Roy. Soc. London A: Math. Phys. Eng. Sci.* 367.1897 (2009), 2429–2445.

[7] Uzma Ali and Punam Khandar. 2013. Data mining for data cloud and compute cloud. *International Journal of Innovative Research in Computer and Communication Engineering* 1, 5 (July 2013), 1137–1141.

[8] Yunhong Gu, Li Lu, Robert Grossman, and Andy Yoo. 2010. Processing massive sized graphs using Sector/Sphere. In *Proceedings of the IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS'10).* IEEE, 1–10.

[9] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and S. I. Spark. 2010. Cluster computing with working sets. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*. USENIX Association Berkeley, CA, 10–10.

[10] Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. 2011. A cloud framework for parameter sweeping data mining applications. In *Proceedings of the IEEE 3rd International Conference on Cloud Computing Technology and Science (CloudCom'11)*. IEEE, 367–374.

[11] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein. 2012. Distributed GraphLab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.* 5, 8 (2012), 716–27.

[12] Aapo Kyrola, Guy E. Blelloch, and Carlos Guestrin. 2012. GraphChi: Large-scale graph computation on just a PC. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*.

[13] Amy Xuyang Tan, Valerie Li Liu, Murat Kantarcioglu, and Bhavani Thuraisingham. 2010. A comparison of approaches for large-scale data mining. Technical Report UTDCS-24-10.

[14] A. Mahendiran, N. Saravanan, N. Venkata Subramanian, and N. Sairam. 2012. Implementation of K-means clustering in cloud computing environment. *Res. J. Appl. Sci. Eng. Technol.* 4, 10 (2012), 1391–1394.

[15] K. Srivastava, R. Shah, D. Valia, and H. Swaminarayan. 2013. Data mining using hierarchical agglomerative clustering algorithm in distributed cloud computing environment. *Int. J. Comput. Theory Eng.* 5, 3 (2013), 520.

[16] Tugdual Sarazin, Mustapha Lebbah, and Hanane Azzag. 2014. Biclustering using Spark-MapReduce. In *Proceedings of the IEEE International Conference on Big Data (BigData'14)*. IEEE, 58–60.

[17] Wei Liu and Ling Chen. 2008. A parallel algorithm for gene expressing data biclustering. *J. Comput. Phys.* 3, 10 (2008), 71–77.

[18] Spiros Papadimitriou and Jimeng Sun. 2008. Disco: Distributed co-clustering with MapReduce: A case study towards petabyte-scale end-to-end mining. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*. IEEE, 512–521.

[19] Esha Sarkar and C. H. Sekhar. 2014. Organizing data in cloud using clustering approach. *Int. J. Sci. Eng. Res.* 5, 5 (2014).

[20] Madhuri H. Parekh. [n.d.]. Enhancement clustering of cloud datasets using improved agglomerative technique. *Int. J. Adv. Netw. Appl.* 128–131.

[21] Renu Ansari. 2015. A distributed k-mean clustering algorithm for cloud data mining. *Int. J. Eng. Trends Technol.* 30, 7 (2015).

[22] Xianfeng Yang and Pengfei Liu. 2013. A new algorithm of the data mining model in cloud computing based on web fuzzy clustering analysis. *J. Theor. Appl. Info. Technol.* 49, 1 (2013).

[23] S. Guha, R. Rastogi, and K. Shim. 1998. June. CURE: An efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, Vol. 27, No. 2. ACM, 73–84.

[24] Madhuri H. Parekh and Ishan K. Rajani. 2015. Improve performance of clustering on cloud datasets using improved agglomerative CURE hierarchical algorithm. *Int. J. Sci. Eng. Technol. Res.* 4, 6 (2015).

[25] Kun Qin, Min Xu, Yi Du, and Shuying Yue. 2008. Cloud model and hierarchical clustering-based spatial data mining method and application. *Int. Arch. Photogram. Remote Sens. Spatial Info. Sci.* 37, B2 (2008), 241–245.

[26] Ran Jin, Chunhai Kou, Ruijuan Liu, and Yefeng Li. 2013. Efficient parallel spectral clustering algorithm design for large data sets under cloud computing environment. *J. Cloud Comput.: Adv. Syst. Appl.* 2, 1 (2013), 18.

[27] Nivranshu Hans, Sana Mahajan, and S. Omkar. 2015. Big data clustering using genetic algorithm on Hadoop MapReduce. *Int. J. Sci. Technol. Res.* 4 (2015).

[28] M. Shindler, A. Wong, and A. W. Meyerson. 2011. Fast and accurate k-means for large datasets. In *Advances in Neural Information Processing Systems*. MIT Press, 2375–2383.

[29] Bhupendra Panchal and R. K. Kapoor. 2013. Performance enhancement of cloud computing with clustering. *Int. J. Eng. Adv. Technol.* 2, 5 (2013).

[30] Pooja Bisht and Kulvinder Singh. 2016. Big data mining: Analysis of genetic K- means algorithm for big data clustering. *Int. J. Adv. Res. Comput. Sci. Software Eng.* 6, 7 (2016).

[31] Alessandro Lulli, Matteo Dell'Amico, Pietro Michiardi, and Laura Ricci. 2016. NG-DBSCAN: Scalable density-based clustering for arbitrary data. *Proc. VLDB Endow.* 10, 3 (2016), 157–168.

[32] Yaobin He, Haoyu Tan, Wuman Luo, Huajian Mao, Di Ma, Shengzhong Feng, and Jianping Fan. 2011. Mr-dbscan: An efficient parallel density-based clustering algorithm using MapReduce. In *Proceedings of the IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS'11)*. IEEE, 473–480.

[33] Dianwei Han, Ankit Agrawal, Wei-Keng Liao, and Alok Choudhary. 2016. A novel scalable DBSCAN algorithm with Spark. In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops*. IEEE, 1393–1402.

[34] F. Ozgur Catak and M. Erdal Balaban. 2012. CloudSVM: Training an SVM classifier in cloud computing systems. In *Proceedings of the Joint International Conference on Pervasive Computing and the Networked World*. Springer, Berlin, 57–68.

[35] Lijuan Zhang and Shuguang Zhao. 2013. The strategy of classification mining based on cloud computing. In *Proceedings of the International Workshop on Cloud Computing and Information Security (CCIS'13)*.

[36] Lijuan Zhou, Hui Wang, and Wenbo Wang. 2012. Parallel implementation of classification algorithms based on cloud computing environment. *TELKOMNIKA Indones. J. Electr. Eng.* 10, 5 (2012), 1087–1092.
[37] Jing Ding and Shanlin Yang. 2012. Classification rules mining model with genetic algorithm in cloud computing. *Int. J. Comput. Appl.* 48, 18 (2012), 24–32.
[38] Jian Wang. 2012. A novel K-NN classification algorithm for privacy preserving in cloud computing. *Res. J. Appl. Sci. Eng. Technol.* 22, 4 (2012), 4865–4870.
[39] Pooja Bajare, Monika Bhoyate, Yogita Bhujbal, Erandole Monika, and Vaishali Shinde. [n.d.]. k-nearest neighbor classification over encrypted cloud data. *IOSR Journal of Computer Engineering (IOSR-JCE).* 45–48.
[40] Apexa B. Kamdar and Jay M. Jagani. 2014. A survey: Classification of huge cloud datasets with efficient map-reduce policy. *International Journal of Engineering Trends and Technology (IJETT)* 18, 2 (2014), 103–107.
[41] Kun Liu and Jan Boehm. 2015. Classification of big point cloud data using cloud computing. *Int. Arch. Photogram. Remote Sens. Spatial Info. Sci.* 40, 3 (2015), 553.
[42] Zhang Danping, Yu Haoran, and Zheng Linyu. 2014. Apriori algorithm research based on MapReduce in cloud computing environments. *Open Autom. Control Syst. J.* 6 (2014), 368–373.
[43] Juan Li, Pallavi Roy, Samee U. Khan, Lizhe Wang, and Yan Bai. 2012. Data mining using clouds: An experimental implementation of Apriori over MapReduce. In *Proceedings of the 12th International Conference on Scalable Computing and Communications (ScalCom'13).* 1–8.
[44] Kuldeep Mishra, Ravi Rai Chaudhary, and Dheresh Soni. 2013. A premeditated CDM algorithm in cloud computing environment for FPM. *Int. J. Comput. Eng. Technol.* 4, 4 (2013), 213–223.
[45] Dheresh Soni, Atish Mishra, and Hitesh Gupta. 2016. An efficient cloud data mining (CDM) algorithm for frequent pattern mining in cloud computing environment. *Lecture Notes Software Eng.* 4, 3 (2016).
[46] Dheresh Soni, Atish Mishra, Satyendra Singh Thakur, and Nishant Chaurasia. 2011. Applying frequent pattern mining in cloud computing environment. *Int. J. Adv. Comput. Res.* 1 (2011), 84–87.
[47] N. Khurana and R. K. Datta. 2013. Pruning large data sets for finding association rule in cloud: CBPA (Count-based Pruning Algorithm). *Int. J. Softw. Web Sci.* (2013), 118–122.
[48] Lijuan Zhou and Xiang Wang. 2014. Research of the FP-growth algorithm based on cloud environments. *J. Software* 9, 3 (2014), 676–683.
[49] Lingjuan Li and Min Zhang. 2011. The strategy of mining association rule based on cloud computing. In *Proceedings of the International Conference on Business Computing and Global Informatization (BCGIN'11).* IEEE, 475–478.
[50] Pooja Godse, Tejal Zete, Mohit Bhanushali, and Shubhangi Kale. 2019. The strategy of mining association rule based on cloud computing. Technical Report. Retrieved 2019 from http://kddlab.zjgsu.edu.cn:7200/research/DistributedMining.
[51] Daniele Apiletti, Elena Baralis, Tania Cerquitelli, Silvia Chiusano, and Luigi Grimaudo. 2013. SeaRum: A cloud-based service for association rule mining. In *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.* 1283–1290. DOI: 10.1109/TrustCom.2013.153
[52] K. Mangayarkkarasi and M. Chidambaram. 2017. An intelligent service recommendation model for service usage pattern discovery in secure cloud computing environment. *J. Theor. Appl. Info. Technol.* 95, 15 (2017).
[53] Daniele Apiletti, Elena Baralis, Tania Cerquitelli, Paolo Garza, Pietro Michiardi, and Fabio Pulvirenti. 2015. PaMPa-HD: A parallel MapReduce-based frequent Pattern miner for high-dimensional data. In *Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW'15).* IEEE, 839–846.
[54] Arkan Al-Hamodi, Songfeng Lu, and Yahya Al-Salhi. 2016. An enhanced frequent pattern growth based on MapReduce for mining association rules. *Int. J. Data Min. Knowl. Manage. Process* 6, 2 (2016), 19–28.
[55] Bo He. 2012. Fast mining algorithm of association rules base on cloud computing. In *Proceedings of the 2nd International Conference on Electronic & Mechanical Engineering and Information Technology.* Atlantis Press.
[56] Wenzheng Zhu and Changhoon Lee. 2014. A new approach to web data mining based on cloud computing. *J. Comput. Sci. Eng.* 8, 4 (2014), 181–186.
[57] R. Farivar et al. 2009. Mithra: Multiple data independent tasks on heterogeneous resource architecture. In *Proceedings of the IEEE International Conference on Cluster Computing and Workshops.* 1–10.
[58] Kyong-Ha Lee, Yoon-Joon Lee, Hyunsik Choi, Yon Dohn Chung, and Bongki Moon. 2012. Parallel data processing with MapReduce: A survey. *ACM SIGMOD Rec.* 40, 4 (2012), 11–20.
[59] Indrajit Roy, Srinath T. V. Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. 2010. Airavat: Security and privacy for MapReduce. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation.* 297–312.
[60] C. Dwork. 2006. Differential privacy. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP'06).*
[61] C. Dwork. 2007. An ad omnia approach to defining and achieving pri-vate data analysis. In *Proceedings of the ACM SIGKDD International Workshop on Privacy, Security, and Trust in Knowledge, Discovery, and Data Mining (PinKDD'07).*

[62] C. Dwork. 2007. Ask a better question, get a better answer: A new approach to private data analysis. In *Proceedings of the International Conference on Database Theory (ICDT'07)*.
[63] C. Dwork. 2008. Differential privacy: A survey of results. In *Proceedings of the International Conference on Theory and Applications of Models of Computation (TAMC'08)*.
[64] Hanna M. Said, Ibrahim El Emary, Bader A. Alyoubi, and Adel A. Alyoubi. [n.d.]. Application of intelligent data mining approach in securing the cloud computing. *Int. J. Adv. Comput. Sci. Appl.* 1, 7, 151–159.
[65] Eric A. Brewer. 2000. Towards robust distributed systems. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC'00)*, Vol. 7.
[66] Werner Vogels. 2008. Eventually consistent. *Queue* 6, 6 (2008), 14–19.
[67] Daniel Abadi. 2012. Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. *Computer* 45, 2 (2012), 37–42.
[68] Domenico Talia. 2013. Toward cloud-based big-data analytics. *IEEE Comput. Sci.* (2013), 98–101.
[69] Robert Grossman and Yunhong Gu. 2008. Data mining using high performance data clouds: Experimental studies using sector and sphere. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 920–927.
[70] Robert L. Grossman, Yunhong Gu, Michael Sabala, and Wanzhi Zhang. 2009. Compute and storage clouds using wide area high performance networks. *Future Gen. Comput. Syst.* 25, 2 (2009), 179–183.
[71] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski, and C. Kozyrakis. 2007. Evaluating MapReduce for multi-core and multiprocessor systems. In *Proceedings of the IEEE 13th International Symposium on High Performance Computer Architecture*. 13–24.
[72] Zhenhua Guo, Geoffrey Fox, and Mo Zhou. 2012. Investigation of data locality in MapReduce. In *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'12)*. IEEE, 419–426.
[73] Domenico Talia and Paolo Trunfio. 2010. How distributed data mining tasks can thrive as knowledge services. *Commun. ACM* 53, 7 (2010), 132–137.
[74] Shivnath Babu. 2010. Towards automatic optimization of MapReduce programs. In *Proceedings of the 1st ACM Symposium on Cloud Computing*. ACM, 137–142.
[75] Eaman Jahani, Michael J. Cafarella, and Christopher R. 2011. Automatic optimization for MapReduce programs. *Proc. VLDB Endow.* 4, 6 (2011), 385–396.
[76] Praveen Kumar Lakkimsetti. 2011. A framework for automatic optimization of MapReduce programs based on job parameter configurations. PhD dissertation, Kansas State University (2011).
[77] Nezih Yigitbasi, Theodore L. Willke, Guangdeng Liao, and Dick Epema. 2013. Towards machine-learning-based auto-tuning of MapReduce. In *Proceedings of the IEEE 21st International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS'13)*. IEEE, 11–20.
[78] Herodotos Herodotou, Harold Lim, Gang Luo, Nedyalko Borisov, Liang Dong, Fatma Bilgen Cetin, and Shivnath Babu. 2011. Starfish: A self-tuning system for big data analytics. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR'11)* 11, 2011 (2011), 261–272.
[79] Vasiliki Kalavri and Vladimir Vlassov. 2013. MapReduce: Limitations, optimizations and open issues. In *Proceedings of the12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'13)*. IEEE, 1031–1038.
[80] Robert Grossman and Yunhong Gu. 2008. Data mining using high performance data clouds: experimental studies using sector and sphere. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 920–927.
[81] F. Ferrucci, P. Salza, M. Kechadi, and F. Sarro. 2015. A parallel genetic algorithms framework based on Hadoop MapReduce. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, 1664–1667.
[82] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery in Databases and Data Mining (KDD'96)*. 226–231.
[83] A. Maithili, R. V. Kumari, and S. Rajamanickam. 2012. Neural networks cum cloud computing approach in diagnosis of cancer. *Int. J. Eng. Res. Appl.* 2, 2 (2012), 428–35.
[84] I. Kaur. 2019. Security of cloud from data mining-based attacks. Technical Report. Retrieved 2019 from https://studyres.com/doc/572585/security-of-cloud-from-data-mining-based-attacks-inderjit.
[85] S. Sharma. 2014. Improving cloud security using data mining. *IOSR J. Comput. Eng.* 1, 16 (2014), 66–69.
[86] Sakshi Aggarwal and Ritu Sindhu. 2014. A survey on cloud mining with privacy protection. *Int. J. Adv. Res. Comput. Sci. Software Eng.* 4, 10 (2014).
[87] Chintada. Srinivasa Rao and Chinta. Chandra Sekhar. 2014. Dynamic massive data storage security challenges in cloud computing environments. *Int. J. Innovat. Res. Comput. Commun. Eng.* 2, 3 (Mar. 2014), ISSN(Online): 2320-9801.
[88] W. Lian, X. Zhu, J. Zhang, and S. Li. 2015. Cloud computing environments parallel data mining policy research. *Int. J. Grid Distrib. Comput.* 8, 4 (2015), 135–144.

[89] Jiong Xie, Shu Yin, and Zhiyang Ding. 2010. Improving MapReduc performance through data placement in heterogeneous clusters. *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'10)*.
[90] A. S. Saabith, E. Sundararajan, and A. A. Bakar. 2016. Parallel implementation of Apriori algorithms on the Hadoop-MapReduce platform—An evaluation of literature. *J. Theor. Appl. Info. Technol.* 85, 3 (2016), 321.
[91] A. A. Pandagale and A. R. Surve. 2016. Hadoop-HBase for finding association rules using Apriori MapReduce algorithm. In *Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT'16)*. IEEE, 795–798.
[92] K. Chandy and L. Lamport. 1985. Distributed snapshots: Determining global states of distributed systems. *ACM Trans. Comput. Syst.* 3, 1 (1985), 63–75.
[93] K. Chandy and J. Misra. 1981. Asynchronous distributed simulation via a sequence of parallel computations. *Commun. ACM* 24, 2 (1981), 198–205.
[94] L. Ismail, M. M. Masud, and L. Khan. 2014. FSBD: A framework for scheduling of big data mining in cloud computing. In *Proceedings of the IEEE International Congress on Big Data (BigData'14)*. IEEE, 514–521.
[95] U. Kang, C. E. Tsourakakis, and C. Faloutsos. 2009. Pegasus: A peta-scale graph mining system implementation and observations. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM'09)*. IEEE, 229–238.
[96] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. 2010. Pregel: A system for large-scale graph processing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 135–146.
[97] Apache giraph. 2019. Retrieved from http://giraph.apache.org.
[98] Giraph. 2019. Retrieved from jira. https://issues.apache.org/jira/browse/GIRAPH.
[99] Avery Ching, Sergey Edunov, Maja Kabiljo, Dionysios Logothetis, and Sambavi Muthukrishnan. 2015. One trillion edges: Graph processing at facebook-scale. *Proc. VLDB Endow.* 8, 12 (2015).
[100] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica. 2013. Graphx: A resilient distributed graph system on spark. In *Proceedings of the 1st International Workshop on Graph Data Management Experiences and Systems*. ACM, 2.
[101] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica. 2014. GraphX: Graph processing in a distributed dataflow framework. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI'14)*. Vol. 14, 599–613.
[102] R. S. Xin, D. Crankshaw, A. Dave, J. E. Gonzalez, M. J. Franklin, and I. Stoica. 2014. Graphx: Unifying data-parallel and graph-parallel analytics. *arXiv preprint arXiv:1402.2394*.
[103] S. Mishra, Y. C. Lee, and A. Nayak. 2016. Distributed genetic algorithm on GraphX. In *Proceedings of the Australasian Joint Conference on Artificial Intelligence*. Springer, 548–554.
[104] E. Y. Chang, H. Bai, and K. Zhu. 2009. Parallel algorithms for mining large-scale rich-media data. In *Proceedings of the 17th ACM International Conference on Multimedia*. ACM, 917–918.
[105] L. Zhou, Z. Zhong, J. Chang, J. Li, J. Z. Huang, and S. Feng. 2010. Balanced parallel fp-growth with MapReduce. In *Proceedings of the IEEE Youth Conference on Information Computing and Telecommunications (YC-ICT'10)*. IEEE, 243–246.
[106] W. Zhang, H. Liao, and N. Zhao. 2008. Research on the FP growth algorithm about association rule mining. In *Proceedings of the International Seminar on Business and Information Management (ISBIM'08)*. IEEE (Vol. 1, pp. 315–318).
[107] I. Pramudiono and M. Kitsuregawa. 2003. Parallel FP-growth on PC cluster. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Berlin, 467–473.
[108] R. Mishra and A. Choubey. 2012. Discovery of frequent patterns from web log data by using FP-growth algorithm for web usage mining. *Int. J. Adv. Res. Comput. Sci. Software Eng.* 2, 9 (2012).
[109] B. S. Kumar and K. V. Rukmani. 2010. Implementation of web usage mining using Apriori and FP growth algorithms. *Int. J. Adv. Netw. Appl.* 1, 06 (2010), 400–404.
[110] Y. Qiu, Y. J. Lan, and Q. S. Xie. 2004. An improved algorithm of mining from FP-tree. In *Proceedings of the International Conference on Machine Learning and Cybernetics*. IEEE, Vol. 3, 1665–1670.
[111] J. Han, J. Pei, and Y. Yin. 2000. Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*. ACM, Vol. 29, No. 2, 1–12.
[112] M. N. Vora. 2011. Hadoop-HBase for large-scale data. In *Proceedings of the International Conference on Computer Science and Network Technology (ICCSNT'11)*. IEEE, (Vol. 1, pp. 601–605).
[113] D. Carstoiu, E. Lepadatu, and M. Gaspar. 2010. Hbase-non SQL database, performances evaluation. *International Journal of Advancements in Computing Technology* 2, 5 (Dec. 2010). DOI: 10.4156/ijact.vol2.issue5.4
[114] S. Nishimura, S. Das, D. Agrawal, and A. El Abbadi. 2011. MD-HBase: A scalable multi-dimensional data infrastructure for location aware services. In *Proceedings of the 12th IEEE International Conference on Mobile Data Management (MDM'11)*. IEEE, Vol. 1, 7–16.

[115] T. Harter, D. Borthakur, S. Dong, A. S. Aiyer, L. Tang, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. 2014. Analysis of HDFS under HBase: A Facebook messages case study. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST'14)*, Vol. 14, 12.
[116] W. Zhao, H. Ma, and Q. He. 2009. Parallel k-means clustering based on MapReduce. In *Proceedings of the IEEE International Conference on Cloud Computing*. Springer, Berlin, 674–679.
[117] R. M. Esteves, R. Pais, and C. Rong. 2011. K-means clustering in the cloud—A Mahout test. In *Proceedings of the IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA'11)*. IEEE, 514–519.
[118] X. Cui, P. Zhu, X. Yang, K. Li, and C. Ji. 2014. Optimized big data K-means clustering using MapReduce. *J. Supercomput.* 70, 3 (2014), 1249–1259.
[119] S. Liu and Y. Cheng. 2012. Research on k-means algorithm based on cloud computing. In *Proceedings of the International Conference on Computer Science & Service System (CSSS'12)*. IEEE, 1762–1765.
[120] T. Sajana, C. S. Rani, and K. V. Narayana. 2016. A survey on clustering techniques for big data mining. *Indian J. Sci. Technol.* 9, 3 (2016).
[121] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic. 2015. Deep learning applications and challenges in big data analytics. *J. Big Data* 2, 1 (2015), 1.
[122] D. Agrawal, S. Das, and A. El Abbadi. 2011. Big data and cloud computing: Current state and future opportunities. In *Proceedings of the 14th International Conference on Extending Database Technology*. ACM, 530–533.
[123] X. Wu, X. Zhu, G. Q. Wu, and W. Ding. 2014. Data mining with big data. *IEEE Trans. Knowl. Data Eng.* 26, 1 (2014), 97–107.
[124] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna. 2013. Cloud-based software platform for big data analytics in smart grids. *Comput. Sci. Eng.* 15, 4 (2013), 38–47.
[125] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos. 2014. Security and privacy for storage and computation in cloud computing. *Info. Sci.* 258 (2014), 371–386.
[126] B. McCarty. 2004. SELinux: NSA's open source security enhanced Linux. O'Reilly Media.
[127] J. Da Silva, C. Giannella, R. Bhargava, H. Kargupta, and M. Klusch. 2005. Distributed data mining and agents. *Int. J. Eng. App. Artific. Intell.* 18, 4 (2005), 791–807. Elsevier Science.
[128] H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson. 2001. Distributed clustering using collective principal component analysis. *Knowl. Info. Syst. J.* 3, 4 (2001), 422–448.
[129] L. Ismail and L. Khan. 2014. *Implementation and Performance Evaluation of a Scheduling Algorithmfor Divisible Load Parallel Applications in a Cloud Computing Environment. Software: Practice and Experience*. Wiley.
[130] M. Shee, S. Bhavsar, and M. Parashar. 1999. Characterizing the performance of dynamic distribution and load-balancing techniques for adaptive grid hierarchies. In *Proceedings of the IASTED International Conference of Parallel and Distributed Computing and Systems*, Vol. 4.
[131] Apache Mahout. 2019. Retrieved from http://mahout.apache.org.
[132] S. Schelter and S. Owen. 2012. Collaborative filtering with apache mahout. In *Proceedings of the ACM RecSys Challenge*.
[133] R. Nair. 2015. Big data needs approximate computing: Technical perspective. *Commun. ACM* 58, 1 (2015), 104–104.
[134] S. Mitra, S. K. Pal, and P. Mitra. 2002. Data mining in soft computing framework: A survey. *IEEE Trans. Neural Netw.* 13, 1 (2002), 3–14.
[135] Foto N. Afrati. 2006. On approximation algorithms for data mining applications. In *Efficient Approximation and Online Algorithms*. Springer, 1–29.
[136] InfoQ. 2019. Approximate Methods for Scalable Data Mining. Retrieved from https://www.infoq.com/presentations/scalability-data-mining.
[137] G. Kollios, D. Gunupulos, N. Koudas, and S. Berchtold. 2001. An efficient approximation scheme for data mining tasks. In *Proceedings of the 17th International Conference on Data Engineering*. IEEE, 453–462.
[138] P. Gupta, S. Agnihotri, and S. Saha. 2013. Approximate data mining using sketches for massive data. *Procedia Technol.* 10 (2013), 781–787.
[139] B. Welton, E. Samanas, and B. P. Miller. 2013. Mr. scan: Extreme scale density-based clustering using a tree-based network of GPGPU nodes. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM, 84.
[140] J. Han and M. Kamber. 2004. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers.
[141] L. Qian, Z. Luo, Y. Du, and L. Guo. 2009. Cloud computing: An overview. In *IEEE International Conference on Cloud Computing*. Springer, 626–631.
[142] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gen. Comput. Syst.* 25, 6 (2009), 599–616.
[143] T. B. Winans and J. S. Brown. 2009. Cloud computing: A collection of working papers. Deloitte LLC.

[144] S. Mittal. 2016. A survey of techniques for approximate computing. *ACM Comput. Surveys* 48, 4 (2016), 62.
[145] J. Gruska. 1999. *Quantum Computing*, Vol. 2005. McGraw-Hill, London.
[146] P. Wittek. 2014. *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Academic Press.
[147] M. Ykhlef. 2011. A quantum swarm evolutionary algorithm for mining association rules in large databases. *J. King Saud Univ.-Comput. Info. Sci.* 23, 1 (2011), 1–6.
[148] S. Wang and G. Long. 2015. Big data and quantum computation. *Chinese Sci. Bull.* 60, 5–6 (2015), 499–508.
[149] P. Rebentrost, M. Mohseni, and S. Lloyd. 2014. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* 113, 13 (2014), 130503.
[150] H. K. Lo, T. Spiller, and S. Popescu. 1998. *Introduction to Quantum Computation and Information*. World Scientific, Singapore.
[151] C. H. Yu, F. Gao, Q. L. Wang, and Q. Y. Wen. 2016. Quantum algorithm for association rules mining. *Phys. Rev. A* 94, 4 (2016), 042311.
[152] D. A. Reed and J. Dongarra. 2015. Exascale computing and big data. *Commun. ACM* 58, 7 (2015), 56–68.
[153] M. Weinstein. 2010. Strange bedfellows: Quantum mechanics and data mining. *Nuclear Phys. B-Proc. Suppl.* 199, 1 (2010), 74–84.
[154] Nature. 2019. IBM's Quantum Cloud Computer Goes Commercial. Retrieved from http://www.nature.com/news/ibm-s-quantum-cloud-computer-goes-commercial-1.21585.
[155] Livemint. 2019. Google's Quantum Computing Push Opens New Front in Cloud Battle. Retrieved from http://www.livemint.com/Technology/FtFrwgaQFFa07m0BenyGIK/Googles-quantum-computing-push-opens-new-front-in-cloud-bat.html.
[156] Engadget. 2019. Google Wants to Sell Quantum Computing in the Cloud. Retrieved from https://www.engadget.com/2017/07/17/google-puts-quantum-computers-to-work-in-cloud/.
[157] Theregister. 2019. Google Tests its Own Quantum Computer – Both Qubits of it. Retrieved from https://www.theregister.co.uk/2016/07/21/google_tests_a_quantum_computer_its_own_both_qubits_of_it/.
[158] Quantum computing – Wikipedia. 2019. Retrieved from https://en.wikipedia.org/wiki/Quantum_computing.
[159] E. Rieffel and W. Polak. 2000. An introduction to quantum computing for non-physicists. *ACM Comput. Surveys* 32, 3 (2000), 300–335.
[160] V. S. Denchev and G. Pandurangan. 2008. Distributed quantum computing: A new frontier in distributed systems or science fiction? *ACM SIGACT News* 39, 3 (2008), 77–95.
[161] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan. 2015. The rise of big data on cloud computing: Review and open research issues. *Info. Syst.* 47 (2015), 98–115.
[162] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J. M. Pierson, and A. V. Vasilakos. 2015. Cloud computing: Survey on energy efficiency. *ACM Comput. Surveys* 47, 2 (2015), 33.
[163] D. Chakrabarti and C. Faloutsos. 2006. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surveys* 38, 1 (2006), 2.
[164] S. Venugopal, R. Buyya, and K. Ramamohanarao. 2006. A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Comput. Surveys* 38, 1 (2006), 3.
[165] I. Goiri, R. Bianchini, S. Nagarakatte, and T. D. Nguyen. 2015. Approxhadoop: Bringing approximations to MapReduce frameworks. In *ACM SIGARCH Computer Architecture News*. ACM, Vol. 43, No. 1, 383–397.
[166] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir. 2014. The rise of RaaS: The resource-as-a-service cloud. *Commun. ACM* 57, 7 (2014), 76–84.
[167] F. Pan, G. Cong, A. K. Tung, J. Yang, and M. J. Zaki. 2003. Carpenter: Finding closed patterns in long biological datasets. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 637–642.
[168] K. A. Shakil and M. Alam. 2016. Recent developments in cloud-based systems: State of art. *Int. J. Comput. Sci. Info. Secur.* 14, 12 (2016), 242.
[169] V. Nekvapil. 2015. Cloud computing in data mining-A survey. *J. Syst. Integr.* 6, 1 (2015), 12.
[170] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob. 2017. Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access* 5 (2017), 5247–5261.
[171] T. Hu, H. Chen, L. Huang, and X. Zhu. 2012. A survey of mass data mining based on cloud-computing. In *Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification (ASID'12)*. IEEE, 1–4.
[172] C. W. Tsai, C. F. Lai, H. C. Chao, and A. V. Vasilakos. 2015. Big data analytics: A survey. *J. Big Data* 2, 1 (2015), 21.
[173] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan. 2014. Big data clustering: A review. In *Proceedings of the International Conference on Computational Science and Its Applications*. Springer, Cham, 707–720.
[174] B. Zerhari, A. A. Lahcen, and S. Mouline. 2015. Big data clustering: Algorithms and challenges. In *Proceedings of the International Conference on Big Data, Cloud and Applications (BDCA'15)*.
[175] A. Mohebi, S. Aghabozorgi, T. Ying Wah, T. Herawan, and R. Yahyapour. 2016. Iterative big data clustering algorithms: A review. *Software: Pract. Exper.* 46, 1 (2016), 107–129.

[176] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras. 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Trans. Emerg. Topics Comput.* 2, 3 (2014), 267–279.
[177] D. Singh and C. K. Reddy. 2015. A survey on platforms for big data analytics. *J. Big Data* 2, 1 (2015), 8.
[178] H. Tong and U. Kang. 2013. Big Data Clustering. *Data Clustering: Algorithms and Applications*, Chapter 11. CRC Press, Taylor & Francis Group, 259–276.
[179] X. Lin. 2014. Mr-Apriori: Association rules algorithm based on MapReduce. In *Proceedings of the 5th IEEE International Conference on Software Engineering and Service Science (ICSESS'14)*. IEEE, 141–144.
[180] Q. He, F. Zhuang, J. Li, and Z. Shi. 2010. Parallel implementation of classification algorithms based on MapReduce. In *Proceedings of the International Conference on Rough Sets and Knowledge Technology*. Springer, Berlin, 655–662.
[181] IBM. 2019. Bluemix is now IBM Cloud. Retrieved from https://www.ibm.com/blogs/bluemix/2017/10/bluemix-is-now-ibm-cloud/.
[182] A. Gheith et al. 2016, IBM Bluemix mobile cloud services. *IBM J. Res. Dev.* 60, 2-3 (Mar. 2016), 7:1–7:12. DOI: 10.1147/JRD.2016.2515422
[183] Google Cloud. 2019. Cloud Machine Learning Engine. Retrieved from https://cloud.google.com/ml-engine/.
[184] GE. 2019. *Predix Platform Brief-GE*. Retrieved from https://www.ge.com/digital/sites/default/files/Predix-The-Industrial-Internet-Platform-Brief.pdf.
[185] TCS. 2019. TCS Connected Universe Platform. Retrieved from https://www.tcs.com/tcs-connected-universe-platform.
[186] IBM Watson | IBM. 2019. Retrieved from https://www.ibm.com/watson/.
[187] Machine Learning Studio | Microsoft Azure. 2019. Retrieved from https://azure.microsoft.com/en-in/services/machine-learning-studio/.
[188] D. R. Krishnan, D. L. Quoc, P. Bhatotia, C. Fetzer, and R. Rodrigues. 2016. Incapprox: A data analytics system for incremental approximate computing. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1133–1144.
[189] Spark Streaming | Apache Spark. 2019. Retrieved from https://spark.apache.org/streaming/.
[190] A. Bifet, S. Maniu, J. Qian, G. Tian, C. He, and W. Fan. 2015. StreamDM: Advanced data mining in Spark streaming. In *Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW'15)*. IEEE, 1608–1611.
[191] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. 2018. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Commun. Surveys Tutor.* 20, 4 (2018), 2923–2960.
[192] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. 2010. Moa: Massive online analysis. *J. Mach. Learn. Res.* 11 (May 2010), 1601–1604.
[193] B. R. Prasad and S. Agarwal. 2016. Stream data mining: Platforms, algorithms, performance evaluators, and research trends. *Int. J. Database Theory Appl.* 9, 9 (2016), 201–218.
[194] G. D. F. Morales and A. Bifet. 2015. SAMOA: Scalable advanced massive online analysis. *J. Mach. Learn. Res.* 16, 1 (2015), 149–153.
[195] A. Amini, T. Y. Wah, and H. Saboohi. 2014. On density-based data streams clustering algorithms: A survey. *J. Comput. Sci. Technol.* 29, 1 (2014), 116–141.
[196] H. Song and J. G. Lee. 2018. RP-DBSCAN: A superfast parallel DBSCAN algorithm based on random partitioning. In *Proceedings of the International Conference on Management of Data*. ACM, 1173–1187.
[197] O. Backhoff and E. Ntoutsi. 2016. Scalable online-offline stream clustering in apache spark. In *Proceedings of the IEEE 16th International Conference on Data Mining Workshops (ICDMW'16)*. IEEE, 37–44.
[198] J. Zgraja and M. Woniak. 2018. Drifted data stream clustering based on ClusTree algorithm. In *Proceedings of the International Conference on Hybrid Artificial Intelligence Systems*. Springer, Cham, 338–349.
[199] C. Sauvanaud, G. Silvestre, M. Kaniche, and K. Kanoun. 2015. Data stream clustering for online anomaly detection in cloud applications. In *Proceedings of the 11th European Dependable Computing Conference (EDCC'15)*. IEEE, 120–131.
[200] L. Tu and Y. Chen. 2009. Stream data clustering based on grid density and attraction. *ACM Trans. Knowl. Discov. Data* 3, 3 (2009), 12.
[201] R. Latif, H. Abbas, S. Latif, and A. Masood. 2015. EVFDT: An enhanced very fast decision tree algorithm for detecting distributed denial of service attack in cloud-assisted wireless body area network. *Mobile Info. Syst.* 2015, Article 260594 (2015), 13 pages. DOI: http://dx.doi.org/10.1155/2015/260594
[202] T. M. Al-Khateeb, M. M. Masud, L. Khan, and B. Thuraisingham. 2012. Cloud guided stream classification using class-based ensemble. In *Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD'12)*. IEEE, 694–701.
[203] J. Chen, K. Li, Z. Tang, K. Bilal, S. Yu, C. Weng, and K. Li. 2017. A parallel random forest algorithm for big data in a spark cloud computing environment. *IEEE Trans. Parallel Distrib. Syst.* 1 (2017), 1–1.

Received December 2017; revised May 2019; accepted July 2019