# Big Data in Data Mining Techniques – A Survey

Dr. S.Sharmila[1], Dr. A.Kanagaraj[2]
[1]*Assistant Professor, Department of Computer Science, NGM College, Pollachi-642001.*
*E-mail: mcasharmi2007@gmail.com*
[2]*Assistant Professor, Department of Computer Science, Kristu Jayanti College*
*Bengaluru-560 077. E-Mail: kanagaraj.a@kristujayanti.com*

***Abstract***

*Big data processing presents itself as a novel and promising analytical field for extracting useful information from enormous databases. It is used to handle vast volumes of knowledge sets, usually large, sparse, incomplete, uncertain, complex, or dynamic information set from various and autonomous sources, in time-sensitive applications such as social site data processing and medical applications. In order for the user to easily obtain the main strategy and answers to their questions from the mined results, massive data processing also handles the storage structure of the mined results. Information slicing is done to break up the associations between columns while keeping the associations within each column. There are several types of information slicing: quasi-static, amorphous, simultaneous dynamic, quasi-static, and dynamic. Another fundamental duty in the huge information mining process is clustering, which is used to find patterns and identify information for use in large-scale processing applications. In addition to discussing the benefits and limitations of these strategies, this study examines huge data processing, information slicing, and clustering techniques. Information slicing and clumping approaches, mining platforms, and large data mining algorithms are discussed along with their quality and performance.*

***Keywords:*** *Big Data Mining, Cloud Computing Technique, Clustering Techniques, Collaborative filtering, Data Mining and Data Slicing.*

## 1. Introduction

Our capacity to collect data in several formats from a wide range of sensors, devices, and independent or linked applications has significantly increased in recent years. The volume of information has increased faster than our ability to handle, evaluate, store, and interpret these datasets. Important obstacles arise while attempting to make use of the vast amount of knowledge. In order to explore vast amounts of data, the mining process looks for consistent patterns or systematic correlations between the variables. The findings are then validated by applying the patterns found to the newly discovered data set. Because of the quality of the information and the increase in computing time, this is frequently an incredibly challenging assignment.

One emerging example that supports the development of connected information centre computer code and infrastructure growth is big information. Massive data could be a computation-focused approach that emphasizes the cloud system's storage capacity. By utilizing massive information processing and storage resources under strict administration, cloud computing technology aims to provide large-scale information applications with fine-grained computing capabilities. As a result, the rise of information also quickens the development of cloud computing, which offers comprehensive options for handling and storing large amounts of data.

Distributed storage technology allows for the efficient management of large amounts of data. Large amounts of data may now be efficiently analysed and stored because to cloud computing's parallel processing capabilities. The application of vast information is currently rapidly expanding across many fields due to the rapid development of networking, information storage, and information sorting capabilities. Large-scale data processing makes it possible to extract useful information from these enormous datasets. Generally speaking, data processing is the process of analysing data from various sources and condensing this data into engaging, comprehensible, and functional models. Higher decision-making requires an appropriate framework for knowledge extraction from databases, as large reservoirs of information gathered from disparate sources demand. Consequently, a hasty decision was made to take advantage of these massive data processing infrastructures.

Massive information and cloud computing technologies are becoming more and more reticular with one another as a result of current technological advancements. Large data runs at a higher level, aided by cloud computing, and offers features akin to information and cost-effective processing power. The growth of cloud computing and the need for applications resulting from virtualized technologies has accelerated the emergence of huge information. Consequently, cloud computing serves as a service mode in addition to offering enormous information processing and calculation capacity. Cloud computing innovations facilitate the big information event, with each technology complementing the other. The primary drivers for the extensive use of cloud computing technology in information technology deployment are its low cost of hardware, low cost of processes, and extensive information testing capability.

Regarding cloud computing technologies, the two most significant concerns are security and managerial loss. In order for the user to easily obtain the main strategy and answers to their questions from the well-mixed results, big data processing also handles the storage structure of well-mixed results. Information slicing is done to break up the associations between columns while keeping the associations within each column. There are several types of information slicing: quasi-static, amorphous, coincidental dynamic, quasi-static, and dynamic.

In the enormous process, lump is also a fundamental task that is carried out for information discovery and pattern extraction for usage in large-scale processing applications. The clump technique includes grid-based, density-based, partition-based, and hierarchical clusters. This paper provides an overview of clump approaches, information slicing techniques, and huge data processing strategies. The merits and downsides of clumping, information slicing, and large data processing approaches are covered in this survey.
The structure of the paper is as follows: The enormous data processing platforms, enormous data processing algorithms, information slicing technique, and clump technique are all illustrated in Section II. The survey's conclusion is presented in Section IV, while Section III details the findings and debates.

## 2. Big Data Mining Technique

This paper provides an overview of bunch approaches, knowledge slicing techniques, and massive data processing strategies. The pros and cons of bunch approaches, knowledge slicing techniques, and massive data processing strategies are covered in this survey.

### 2.1 Huge Data Processing Platforms

### A) MapReduce

A distributed programming model called MapReduce [1] may be used for large clusters of systems that will process large datasets in parallel. The responsibility for managing the Map and cutback approach is with the work hunter. The outputs of the maps are sorted by the MapReduce framework [2], and these are subsequently used as input for the reduction tasks. The work area unit's input and output are all kept inside the classification scheme. The analysis community has given the MapReduce model, which was first introduced by Google, significant attention as parallelizing data processing algorithms have been used by the model due to its parallel computing nature. Writing down the queries in Python or Java is really laborious. The map section must be finished before the cutback section can start. The considerable performance reduction is the result of this.

### B) HBase

One column-based management system that is frequently utilized for large knowledge applications is Hbase [3]. Actually, aggregating the various properties into column families and storing each member of the column family together is made possible by HBase [4]. Hbase may be a distributed, climbable database for arbitrary read/write operations as well as holding massive tables. It also enables fault-tolerant storing of large amounts of distributed knowledge with quick access. But storing massively large binary files in HBase is really difficult. In terms of memory block allocations and hardware requirements, HBase is expensive.

### C) Dynamo Amazon

A completely controlled information system that supports key-value and document knowledge models is DynamoDB [5]. In order to facilitate several tiers of request flow and store and retrieve any amount of knowledge, Generator [6] generates an information table. The information table will automatically connect to a sufficient number of servers in order to handle the client's requested capability level and data storage capacity while preserving consistency and a rapid performance rate. But it only supports a limited set of questions, therefore determining knowledge consistency is more difficult. The categorization field in Dynamo DB must be set before the information is created and cannot be altered after that.

### D) Asterix

The goal of Asterix [7] is to combine the robust concepts of Web-scale computing with those of parallel information systems, such as fault tolerance for lengthy tasks. Asterix [8] is intended to be a parallel, semi-structured information management system that can absorb, store, index, query, analyze, and publish large amounts of semi-structured data. Asteroid is an excellent tool for handling exceedingly complex, diverse, and rigid knowledge. Both external and system-managed datasets are supported. The lack of a cost-based query optimizer is the main drawback of the asteroid.

### E) Hadoop

At first, Hadoop [9] is thought to support MapReduce, just like Java is supported for programming languages. In the Hadoop platform [10], the application is broken down into numerous little pieces. A generic execution engine that parallelizes computation over an enormous cluster of machines is the MapReduce programming model, which is used by Hadoop to develop applications that quickly multiprocess large amounts of data on massive clusters. Given that Hadoop offers inexpensive and dependable storage, it will provide a distributed system with the necessary strength and choice for measurability. Hadoop isn't the easiest solution, though, for businesses that deal with knowledge. Within the Hadoop platform, there are issues with multi-tenancy and cascade failure.

## 2.2 Massive Data Processing Algorithms

Algorithms for big data processing [11] are used to find information or identify patterns in large amounts of data. Many algorithms are employed in large-scale data processing units. The fuzzy C-Means (FCM) algorithmic program, the two-section top-down specialization (TPTDS) technique, the tree-based Association Rules (TARs), and the Associate Rule Mining (ARM) algorithmic program.

### A) Two-Phase top-down Specialization (TPTDS) approach

The TPTDS methodology [12] is specifically employed for very private knowledge mining. This technique is divided into two phases: task level and job level. All the phases achieve parallelization. The main goal of TPTDS [13] is to establish a relationship between information usefulness and quantifiability in order to obtain high quantifiability. The main drawback of the TPTDS approach, despite its widespread application for privacy preservation on highly sensitive large-scale information sets, is that it is unable to provide privacy preservation for information sets with enormous quantability.

### B) Tree-Based Association Rule (TAR)

The extensile terminology (XML) documents are mined using Tree-Based Association Rule [14], and as a result, the results are frequently stored in XML forms. In a vast amount of gathered data, co-occurrence items are described by association rules [15]. Support and confidence are used to gauge the Associate in Nursing Association rule's standard. To accommodate the ranking character of XML documents, the association rule is expanded in the context of relative databases. It is common to find relationships between XML document subtrees with regard to the matter contents of leaf components and attributes pricing. The capacity to update the TAR-storing document during an amendment occurs in the original XML information sets, and the Tree-Based Association Rule may be limited by their index.

### D) Fuzzy C-means (FCM)

The FCM algorithmic software [16] is utilized for the mining of labeled data, big picture data, and unloadable data from a cluster of extremely large data. Two goals, namely acceleration for loadable information and approximation for unloadable information, will be achieved by the cluster techniques. Compared to crisp partitions, fuzzy partitions are more flexible because each object can belong to more than one cluster. In order to operate as the initial cluster centers, FCM [17] is initialized by randomly selecting objects from the dataset. The algorithmic program ends when there are only minimal changes in the cluster center positions. The algorithmic program known as fuzzy C-Means has numerous difficulties when it comes to creating and researching ascendible solutions for really large fuzzy clusters. It may be possible to choose the best kernel by using cluster validity indices to examine areas where kernel solutions are frequently employed. The requirement for complete access to the object vector information serves as a gauge for cluster quality.

### E) Associate Rule Mining (ARM)

ARM [18] has the potential to be a useful method for clinical data processing applications by exposing significant relationships between variables in information sets. ARM [19] is widely used in hospital diagnostics for medical specialties, aid auditing, and cardiac condition prediction, among other applications. Two essential measures are confidence and support, which evaluate the degree of association and frequency of a rule. The mining approach requires the users to define minimal support and minimum confidence values as thresholds in order to find frequent and assured association rules. Finding all frequent and reliable rules that support these two users' nominative values is ARM's primary objective. The manual definition of variables for interest and cut points by doctors is the main ARM restriction. Table.1 shows the comparison of the massive data processing algorithms.

**Table 1.** Comparison of Big Data Mining Algorithms

| Algorithm/ Approach | Performance Criteria | Usage |
|---|---|---|
| Two-Phase Top-Down Specialization (TPTDS) approach | Execution time and Information Loss | Privacy Preservation Of data |
| Tree-Based Association Rules (TAR) approach | Extraction time and Answer time | Mining from semi structured (XML) document |
| FCM Algorithm | Run time | Clustering of data |
| Associate Rule Mining (ARM) | Comorbidity | Mining from ICU(clinical) data |

## 2.3 Information Slicing Technique

Data slicing [20] breaks the associations between columns, but keeps the associations within each column. Slicing divides the dataset both vertically and horizontally. In doing so, the greater information utility level is preserved while the information spatiality is decreased. Both horizontal and vertical dataset partitioning are included in information slicing. Grouping the attributes into columns that support the correlations between the attributes completes the vertical partitioning process. Each column has a collection of characteristics that exhibit strong correlations. The final step in horizontal partitioning is to organize tuples into buckets. Finally, values in each column are randomly permuted inside each bucket to break the association between entirely separate columns. There are several types of information slicing: quasi-static, coincident dynamic, amorphous, quasi-static, and dynamic.

### A) Static Slicing

Static slicing [21] is carried out without making any assumptions about the program's input. For computing static slices, dependencies between entirely distinct field components of the unified modeling language (UML) model are known. Starting from the slicing criterion, a backward traversal of the program's management flow graph (CFG) or program dependence graph (PDG) is used to compute the slices [22]. This process gathers statements and management predicates. Nevertheless, a static slice could include statements that don't affect the values of the relevant variables during the execution. A program's execution may result in unexpected outcomes depending on the value entered.

### B) Dynamic Slicing

Using dynamic slicing [23], the program is fed information while it is being executed, and as a result, the slice only includes the statement that failed during the specific execution of interest. Using dynamic analysis, dynamic slicing [24] finds all and only those statements on the real aberrant execution trace that affect the variables of interest. While static slicing views each definition or use of an array component as a definition or use of the entire array, dynamic slicing is able to handle each component of the Associate in nursing array individually. But dynamic slice is also enormous for a reasonable large software application. The dynamic slicing method yields workable slices that are accurate on a single input.

### C) Simultaneous Dynamic Slicing

The union of the dynamic slices on the element test cases does not provide simultaneous dynamic slicing [25] on a set of test cases. In fact, the union of dynamic slices alone is flawed since it fails to maintain coincident accuracy over all inputs. A repetitive formula for nursing is provided, which computes a larger dynamic slice with each iteration, starting from the initial collection of statements and gradually building the coinciding dynamic slice. This method will be used in program comprehension to isolate a group of statements, such as particular program activity. Because coinciding dynamic slicing considers the program's information flow and enables the reduction of the list of selected statements, it will be considered an improvement over current approaches for localizing functions supported by look-cases.

### D) Quasi-Static Slicing

A combination of static and dynamic slicing is known as quasi-static slicing [26]. Some variables is mounted in quasi-static slicing [27], and as a result, the program is examined throughout variations in the values of other variables. Regarding the slicing criterion, the first program's behavior remains unchanged. Similar slicing is known as Conditioned slicing since the slicing criteria incorporates the collection of variables of interest and initial circumstances. This may be the cost-effective approach for understanding the Associate in nursing degree. However, the uniform treatment of static and dynamic slicing is not demonstrated by this method.

### E) Amorphous Slicing

Like the slicing criterion, amorphous slicing [28] depends on safeguarding the program's language. The varied slicing methods appear to have reduced the size of the fashioned slices.

The program is notably simplified in the slice when considering the slicing requirement. Comprehension, analysis, and reprocessing of programs are aided by amorphous slicing. However, language is maintained by using the continuous folding technique to realize the slicing once amorphous slicing is applied to the variable's ultimate worth. Instead of eliminating offensive sentences, the amorphous slicing accomplishes a manageable syntactic alteration while maintaining the linguistics in respect to the slicing requirement. This slice's output is never allowed to exceed the size of the first program to be sliced.

### 2.4 bunch technique

The fundamental aim of the information mining approach is to cluster data, or group similar pieces of information together in order to reduce the amount of data. This is known as the clustering process [29]. A multitude of algorithms are employed for group. Among the most popular bunch procedures are methods based on grids, partitioning, hierarchies, and densities.

#### A) Hierarchical cluster methodology

By combining smaller clusters into larger ones or breaking up the larger clusters, the hierarchical cluster [30] constructs the clusters gradually. The result of the algorithmic rule is a cluster tree, or dendrogram, that illustrates the relationships between the clusters. By slicing the dendrogram at a specified level, the information items were clustered into distinct teams. Agglomerate stratified clusters and discordant stratified clusters are two broad categories for stratified clusters. Every information point in the aggregate approach is regarded as a distinct cluster, and as a result, the clusters are unified on all iteration-supported criteria. All data points are viewed as a single cluster in the discordant approach and are scattered into several clusters based on specific criteria. The inability to make modifications during the splitting/merging process, the lack of interpretability with regard to the cluster descriptors, and the imprecise termination criteria are the main drawbacks of the stratified cluster methodology. The curse of spatial property development causes severe effectiveness degradation in high dimensional areas.

#### B) Partitioning cluster methodology

A collection of disjoint clusters is created by directly breaking down an information set using the partitioning cluster methodology [31]. The primary drawback of clusters is the crucial choice about the number of clusters and the emergence of diverse cluster types. It may also be important to format the cluster's centroids. If a cluster's centroids are positioned far from the information distribution, some clusters may remain empty. In order to overcome these constraints, information objects are divided using the partitioning technique into multiple partitions, each of which represents a cluster. Severe reduction in efficiency in high dimensional regions since most point pairs are about as far apart as the average. In high dimensional environments, the concept of distance between points is not well defined. The partitioning cluster methodology allows for the modification of non-convex clusters with varying sizes and densities and is a very intelligent way to format sections, noise, and outliers.

#### C) Density-based cluster methodology

Information items are divided using the density-based cluster technique [32], which also supports density regions, properties, and boundaries. A connected dense element continues to expand within the specified cluster according to a density-based algorithmic rule as long as the neighborhood's density surpasses a predetermined intensity level. As a result, the density-based algorithms will find the erratic form clusters and provide noise and outlier protection. As a result, the density function is computed for the dataset functions that determine a particular piece of information. This approach's inadequate cluster descriptors and extreme sensitivity to input parameter settings are its biggest drawbacks.

**D) Grid-based cluster methodology**

In order to carry out the cluster operations, the object house is divided into a limited number of cells that form a grid structure using the grid-based cluster approach [33]. This method does not depend on the number of information items; rather, it depends only on the number of cells in each dimension inside the amount house. This method, which is primarily grid-based, walks through the dataset to get the applicable math value for the grids. As a result, the grid-based, largely cluster-based approach's interval is quick. Because the grid's dimensions are far smaller than the information's dimensions, the grid-based methodology performs better. However, for severely irregular information distributions, using a single uniform grid isn't decent to attain the necessary cluster quality.

## 3. Results and Discussions

This section illustrates several methods for handling huge processes. This section discusses the performance and quality evaluation of the mining platforms, cluster methodologies, information slicing strategies, and large data processing algorithms.

**Table 2. Information about various techniques involved in Big Data Mining Process.**

| Techniques | Author& Reference | Year | Performance | Quality Measurement |
|---|---|---|---|---|
| | | **Big Data Mining Platforms** | | |
| **Map Reduce** | J.Li et al [1] | 2012 | The key benefit of MapReduce is that it automatically handles failures, hiding the complexity of fault-tolerance from the programmer. | 1. Co-scheduling speedup<br>2. Execution time<br>3. Number of processors |
| | H.Wang et al [2] | 2012 | This new approach improves the ability to scale up the Hadoop clusters to a much larger configuration. In addition to this, it also permits parallel execution of a range of programming models. | 1. Feature extraction time<br>2. Clustering time<br>3. Average accuracy<br>1. Mean average precision |
| **Hbase** | Aksu et al [3] | 2013 | Hbase supports random and fast insert, update and delete access. | 1. K-core construction time<br>2. Execution time<br>3. Maintenance overhead |
| | Rabl et al [4] | 2012 | Hbase provides linear and modular scalability, strictly consistent data access, automatic and configurable sharing of | 1. Throughput<br>2. Read latency<br>3. Write latency |

| | | | | |
|---|---|---|---|---|
| | | | data. | |
| **Dynamo** | R.Gupta et al [5] | 2012 | Dynamo provides incremental scalability. Hence, keys are partitioned dynamically using a hash function to distribute the data over a set of machines or nodes. | - |
| | Moharil et al [6] | 2014 | Dynamo provides faster and predictable performance with seamless scalability with minimal database administration. | 1. Number of executors and tasks<br>2. Processing time<br>3. Lines tailed<br>4. Number of clusters formed |
| **Asterix** | Behm et al [7] | 2011 | Asterix supports large, self-managing data sets and index structures as well as query planning, processing, and scheduling approaches that are scalable and adaptable to highly dynamic resource environments. | 1. Speedup Ratio<br>2. Number of nodes in cluster<br>3. Training size<br>4. Loss on test data |
| | Kaldewey et al [8] | 2012 | Since ASTERIX provides rich spatial support, spatial aggregation queries are executed efficiently by using a secondary R-tree index. Thus, all records outside of the query bounding region are filtered quickly. | 1. Hadoop distributed file system (HDFS) read bandwidth<br>2. Disk bandwidth |
| **Hadoop** | Dede et al [9] | 2013 | The design of Hadoop achieves data locality and considerable performance improvement by placing data on the compute nodes. | 1. Check point interval<br>2. Overhead<br>3. Number of tasks<br>4. Number of input records<br>5. Processing time |
| | Zhang et al [10] | 2013 | Hadoop can effectively reduce the search time and improve the retrieval speed. | 2. Data Capacity<br>3. Cluster Processing time<br>4. Single node cost time<br>4. Cluster cost time |
| **Big Data Mining Algorithms** | | | | |

| | | | | |
|---|---|---|---|---|
| **Two-Phase Top-Down Specialization (TPTDS) approach** | X.Zhang et al [12] | 2014 | The two-phase TDS approach gains high scalability via allowing specializations to be conducted on multiple data partitions in parallel. | 1. Execution time<br>2. Information loss |
| | X.Zhang et al [13] | 2014 | It gains high scalability and efficiency of sub-tree anonymization scheme to anonymize data sets for privacy preservation. | 1. Privacy preserving cost<br>2. Privacy leakage degree<br>3. Number of datasets |
| **Tree-Based Association Rule (TAR)** | A. R. Islamand T.S.Chung[14] | 2011 | Tree–Based Association Rules performs mining all frequent association rules without imposing any prior restriction on the structure and the content of the rules. | 1. Extraction time<br>2. Answer Time |
| | K.S. Rani et al [15] | 2013 | Enables the user to extract efficient answering from the XML documents. | 1. Number of cluster nodes<br>2. Incremental eventuate size<br>3. Execution time |
| **Fuzzy C-means(FCM)** | S.Kannan et al [16] | 2012 | The Fuzzy C-Means algorithm supports the clustering of very large data or big data. | 1. Runtime |
| | H. Izakian and A. Abraham[17] | 2011 | The Fuzzy C-means algorithm is efficient, straightforward and easy to implement. | 1. Number of clusters |
| **Associate Rule Mining (ARM)** | F. Suchanek and G. Weikum [18] | 2013 | Associate Rule Mining algorithm generates quantitative and real time decision support rules for Intensive Care Unit (ICU) by predicting the characteristics of ICU stay. | 1. Comorbidity |
| | Galárraga et al [19] | 2013 | The associate rule mining algorithm achieves improved run time, quality of the output rules and reasonably predicts the precision of the rules. | 1. Aggregated predictions<br>2. Aggregated precision |
| **Data Slicing** | | | | |
| **Static Slicing** | Alomari et al [21] | 2012 | The approach is highly scalable and can generate the slices for all variables of the Linux kernel in less than 13 minutes. | 1. Slice size<br>2. System size |

|  |  |  |  |  |
|---|---|---|---|---|
|  | Santelices et al [22] | 2013 | The accuracy of slices is improved. Slicing can be done within a short period. | 1. Percentage of slice inspected<br>2. Percentage of impacts found<br>3. Slicing ranking<br>4. 4. Run time overhead |
| **Dynamic Slicing** | J. T. Lallchandani and R. Mall[23] | 2011 | The advantage of dynamic slicing isthe run-time handling of arrays and pointer variables. | 1. Reverse execution time<br>2. Selection sort range<br>3. Average speedup<br>4. Average code size reduction<br>5. Memory overhead |
|  | J. Zhong and B. He[24] | 2014 | Dynamic slicing produces a more compact and precise slice. Reverse execution along a dynamic slice skips recovery of unnecessary program state. | 1. Sliced execution time<br>2. Slice size<br>3. Workload<br>4. Probability |
| **Simultaneous Dynamic Slicing** | M. A. El-Zawawy [25] | 2014 | As the slice size is dynamically calculated, the lines of code do not affect the slice criteria. Hence, simultaneous dynamic slicing achieves better performance under all the conditions. The time required for calculating the slice size is reduced, since the slice size is calculated on the runtime. | 1. Lines of code<br>2. Slice size<br>3. Distance<br>4. Percentage affect<br>5. Function points |
| **Quasi-Static Slicing** | Swain et al [26] | 2012 | It allows a better decomposition of the program giving the maintainer the possibility to analyze code fragments with respect to different perspectives. | 1. Number of test cases<br>2. Slice test coverage |
|  | S. Koushik and R. Selvarani[27] | 2012 | Quasi-static slicing allows a better decomposition of the program giving human readers the possibility to analyze code fragments with respect to different perspectives. | 1. Execution time<br>2. Performance gain<br>3. Bounded model checking (BMC) |
| **Amorphous Slicing** | Androutsopouloet al [28] | 2013 | It produces slice of smaller size. | 1. Slice size<br>2. Number of lines of codes for slice<br>3. Execution time |
| **Clustering Technique** |  |  |  |  |

| | | | | |
|---|---|---|---|---|
| **Hierarchical Clustering Method** | M.Verma et al [30] | 2012 | Hierarchical clustering method is more versatile and easy to handle any forms of similarity or distance. It is consequently applicable to any attribute types. | 1. Number of clusters<br>2. Cluster instances |
| **Partitioning clustering Method** | K. Aparna and M. K. Nair[31] | 2015 | The partition-based technique utilizes an iterative way to create the cluster. | 1. Number of clusters<br>2. Data points |
| **Density-based Clustering Method** | Kriegel et al [32] | 2011 | This approach enables discovery of arbitrary-shaped clusters with varying size. It is more resistance to noise and outliers. | 1. Number of clusters<br>2. Cluster instances<br>3. Number of iterations<br>4. Time taken to build model<br>5. Log likelihood |
| **Grid-based Clustering Method** | E. G. Mansoori [33] | 2014 | The grid-based clustering technique requires low processing time, since it depends on the size of the grid instead of the size of the data. | 1. Time consumption<br>2. Clustering correct rate<br>3. Noise filtering rate |

## 4. CONCLUSION

Big data processing also handles the well-mixed result storage structure, which makes it easy for the user to obtain the main strategy and query answers from the well-mixed result set. Knowledge slicing is used to break associations between columns while maintaining associations within each column. In the huge data {processing} process, cluster is also a fundamental activity carried out for information discovery and obtaining patterns to be utilized in the massive processing applications. The large-scale data processing methods, mining platforms, knowledge slicing strategies, and cluster strategies are discussed together with their performance and quality.

## References

[1] J. Li, P. Roy, S. U. Khan, L. Wang, and Y. Bai, "Data mining using clouds: An experimental implementation of apriori over mapreduce," in 12th International Conference on Scalable Computing and Communications (ScalCom),2012.

[2] H. Wang, Y. Shen, L. Wang, K. Zhufeng, W. Wang, and C. Cheng, "Large-scale multimedia data mining using MapReduce framework," in CloudCom, 2012, pp.287-292.

[3] H. Aksu, M. Canim, Y.-C. Chang, I. Korpeoglu, and O. Ulusoy, "Multi-resolution Social Network Community Identification and Maintenance on Big Data Platform," in IEEE International Congress on Big Data (BigData Congress), pp. 102-109,2013.

[4] T. Rabl, S. Gómez-Villamor, M. Sadoghi, V. Muntés-Mulero, H.-A. Jacobsen, and S. Mankovskii, "Solving big data challenges for enterprise application performance management," Proceedings of the VLDB Endowment, vol. 5, pp. 1724-1735,2012.

[5] Dr. P.Logeswari “Extraction of Subset- Want in Data Stream using EMDMICA Algorithm “ Volume 7 Issue VI, June 2019.

[6] Dr. P.Logeswari, J.Gokulapriya “A Literature Review on Data Mining Techniques “in July Volume -7 Issue -7.

[7] .Dr. P.Logeswari, J.Gokulapriya “Literature Survey on Big Data mining And Its Algorithmic Techniques “in July Volume -8 Issue7.

[8] Dr. P.Logeswari, G.Banupriya “A Survey on Implementations Solutions for Attack Prevention Cryptography Technique’s in WSN UsingNS2” Volume 7,Issue 6 June 2021.

[9] Dr. P.Logeswari, G.Banupriya “Review on Cryptography Techniques in WSN for Attack Prevention” volume 8, Issue 8.

[10] Dr. P.Logeswari, S.Sudha “A Survey on Privacy Preserving in Data Mining”Volume-7, Issue-8 August 2021.

[11] Dr. P.Logeswari, S.Sudha “A Review on Privacy Preserving in Data Mining” Volume-8, Issue-6 June2021.

[12] Sangeetha, J. and Prakash, V.S., 2017. A survey on big data mining techniques. International Journal of Computer Science and Information Security, 15(1), p.482.

[13] Yang, J., Li, Y., Liu, Q., Li, L., Feng, A., Wang, T., Zheng, S., Xu, A. and Lyu, J., 2020. Brief introduction of medical database and data mining technology in big data era. Journal of Evidence-Based Medicine, 13(1), pp.57-69.

[14] Hussan, M.I.T., Reddy, G.V., Anitha, P.T. et al. DDoS attack detection in IoT environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization. Cluster Comput (2023).

[15] Nti, I.K., Quarcoo, J.A., Aning, J. and Fosu, G.K., 2022. A mini-review of machine learning in big data analytics: Applications, challenges, and prospects. Big Data Mining and Analytics, 5(2), pp.81-97.

[16] George, A.H., Shahul, A., George, A.S., Baskar, T. and Hameed, A.S., 2023. A Survey Study on Big Data Analytics to Predict Diabetes Diseases Using Supervised Classification Methods. Partners Universal International Innovation Journal, 1(1), pp.1-8.

[17] Joseph Gladju, Ayyasamy Kanagaraj, Kamalam Biju Sam, Use of data mining to establish associations between Indian marine fish catch and environmental data, Archives of Biological Sciences, Vol. 75 No. 4 (2023),pp. 459-474.

[18] J Gladju, BS Kamalam, A Kanagaraj,(2022), Applications of data mining and machine learning framework in aquaculture and fisheries: A review, Smart Agricultural Technology,vol. 2. pp.1-15.