https://doi.org/10.31449/inf.v47i6.4445 Informatica **47** (2023) 203–218 **203**

Big Data Clustering Techniques Challenges and Perspectives: Review

Fouad H. Awad[1], Murtadha M. Hamad[1]
[1]College of Computer Science and Information Technology, University of Anbar, Anbar, Iraq
E-mail: fouad.hammadi@uoanbar.edu.iq, co.mortadha61@uoanbar.edu.iq

Keywords: Big data, clustering, data mining, machine learning

Received: October 12, 2022

Clustering in big data is considered a critical data mining and analysis technique. There are issues with adapting clustering algorithms to large amounts of data and new challenges brought by big data. As the size of big data is up to petabytes of data, and clustering methods have high processing costs, the challenge is how to handle this issue and utilize clustering techniques for big data efficiently. This study aims to investigate the recent advancement of clustering platforms and techniques to handle big data issues, from the early suggested techniques to today's novel solutions. The methodology and specific issues for building an effective clustering mechanism are presented and evaluated, followed by a discussion of the choices for enhancing clustering algorithms. A brief literature review of the recent advancement in clustering techniques has been presented to address each solution's main characteristics and drawbacks.

Povzetek: Članek predstavlja pregled tehnik gručenja za velike podatke.

1 Introduction

After a period of addressing challenges associated with processing big data, the emphasis has shifted towards making sense of this vast volume of information. According to experts and scholars, big data represents one of the most pressing concerns in the field of computer science today. A notable example of the scale of big data can be seen in YouTube, which boasts one billion daily active users who collectively upload 100 hours of content every hour. Additionally, its copyrighted content service must evaluate over 400 years of video content on a daily basis [31]. A billion users on Facebook and Twitter are creating terabytes of data every minute. It is critical to employ advanced knowledge discovery tools to deal with this flood of data. Data mining techniques [68] [54] are excellent information-seeking tools for this purpose. One of them is Clustering, which is described as "a strategy for dividing data into groups in such a way that items in one group have more similarity than objects in other groups" [31]. Data clustering is a well-known strategy in many parts of computer science and related disciplines. Although data mining is the most widely used method of Clustering, it is also commonly used in other subject areas such as biostatistics, energy studies, deep learning [37], networking, and pattern classification [1] [5], resulting in a large body of research [41]. Researchers have studied clustering algorithms since their inception to regulate their complexity and processing cost and, consequently, improve scalability and performance. Big data's rise in popularity in recent years has introduced new challenges to this field, prompting further research into improved clustering methods. It is critical to determine the size of the data before focusing on clustering it. Bezdek and Hathaway established a data size classification to solve this challenge [50].

There are five important challenges of big data, which are [24]:

- Volume: it is the first, exemplified by streaming unstructured data in the form of social media, which raises questions of how to assess the relevant data within high data volumes and how to provide helpful information by evaluating data.
- Velocity: Data is pouring at breakneck performance, and it must be processed in a fair amount of time. One of the issues of big data is responding fast to data velocity.
- Variety: Another problematic challenge is merging, managing, and evaluating data from several sources with varying standards, such as social media posts, audio, unstructured data, and video.
- Veracity: This refers to the data's quality, and it shows the data's biases, noise, and abnormalities.
- Value: This points to the precious knowledge revealed from the data.

Due to their massive complexity and processing cost, traditional clustering approaches cannot handle this enormous volume of data. In the standard K-means algorithm, for example, even when the number of clusters is 2, the NP-hard. As a result, scalability is the most challenging aspect of massive clustering data[35]. The scaling and speed-up clustering algorithm is the crucial goal while sacrificing as little quality as possible. Although researchers in this field have always sought to improve the scalability and performance of clustering algorithms, big data concerns highlight these flaws and demand additional attention and research.

Figure 1: Big data challenges

Because of their incredible complexity and processing cost, traditional clustering approaches cannot handle this massive volume of data. Traditional Kmeans clustering, for example, is NP-hard even when k=2, which is the number of clusters. As a result, scalability is the most challenging aspect of massive clustering data[35].

Scaling and speeding up are the critical goals of clustering algorithms while sacrificing as little quality as possible. Although researchers in this field have always sought to improve the scalability and speed of clustering algorithms, significant data concerns highlight these flaws and demand additional attention and research. According to a review of the literature on clustering approaches, these techniques have progressed through five stages, as shown in Figure 1 [76].

Many papers have been discussed and written to be review papers for the clustering techniques in big data due to the importance of those types of studies to understanding Clustering and start working with it. However, compared with the recent review papers in big data clustering [60] [33] [75] [23], this study has many key differences that do not present in the previous review papers which represent the novelty as follows:

- This review focuses only on the recent clustering techniques and algorithms used for the modern big data processing frameworks after reviewing more than 200 papers.
- Addressing the main issues of clustering technique related to big data.
- Representing the main characteristics and differences between the most used big data platform which is Apache Spark and Hadoop MapReduce, which may guide the researchers to select the based platform based on the issue and requirements of the big data system
- Filtering the most advanced clustering solution in the big data field with addressing each solution's main advantages and disadvantages.

In the rest of this research, the drawbacks and benefits of algorithms in each step have been studied in the sequence they occur in the diagram. Finally, based on current and unique approaches and future work, an additional stage has been revealed that might be the next stage for large data clustering algorithms. Techniques for enabling clustering mechanisms to operate with larger datasets by increasing the performance and scalability are divided into two sections:

- Techniques for clustering single machines
- Techniques for clustering many machines

Single-machine clustering algorithms run on a single system and can only use that machine's resources, whereas multiple-machine clustering techniques are accessed to resources and run on several machines. In the sections below, further information will be discussed about those techniques[30].

2 Survey methodology

We examined the most relevant research publications published between 2015 and 2022, mainly from 2021 and 2022, with a few from 2019. Papers from reputable publications such as IEEE, Elsevier, MDPI, Nature, ACM, and Springer were the primary emphasis, and some papers were chosen from prestigious conferences. We looked at over 200 papers on various big data clustering topics. Seventy-five papers from the year 2022, 60 papers from year 2021, and 30 papers from the year 2019 are included in this collection. That means that this study focused on recent articles in the field of big data clustering. The publications were evaluated and reviewed to (1) define and list big data methods, and clustering algorithms, (2) describe and explain big data structures, and (3) offer clustering issues and alternative solutions. ("Big Data"), ("Clustering Techniques"), ("Clustering Platform"), ("single-machine Clustering," AND "multi-machine Clustering") are the most often utilized keywords for this review paper's search criteria.

3 Big data

"Big Data" describes a set of algorithms, methods, and technologies designed to handle large amounts of data. Greater storage capacity, enhanced data storage, improved processing capabilities of high-performance computers, and access to extensive data have all supported the growth of the Big Data Processing industry. Hardware and software in today's world can manage, alter, analyze, and analyze massive amounts of data in a novel manner. The proliferation

of the Internet, social media, smartphones, and apps has resulted in a data tsunami. Collecting and analyzing enormous volumes of data can reveal patterns, trends, and correlations relating to human actions and interactions. Big data has been used to analyze customer behavior, target marketing efforts, improve tasks, improve efficiency, and minimize risk. According to IDC, a global source of industry insight and information technology consulting services, in the subsequent years, the global analytics of big data is expected to expand rapidly [7]. Organizations need help to figure out how to make the most of this plethora of data. There are three types of data: identity, intelligence, and people. Because of the size of big data, typical processing techniques are frequently insufficient [46]. Big Data is listed as a future technology in Gartner's 2014 Hype Cycle.

Because data is continually created and we are inundated with it, what today considers 'big' tomorrow may not be 'big.' As a result, traditional data processing approaches that can not scale to large data will eventually become outdated. Using multiple Big Data processing frameworks, each client can handle Big Data and strive to extract value from it. Hadoop and Spark are two of the most prominent big data processing open-source frameworks, while others have better specialized in their use yet have nonetheless managed to gain reputations and good market sharing. [**?**]. In general, there are two types of frameworks: proprietary and open-source, both widely utilized in the big data field.

4 Big data clustering

There are two types of clustering techniques in big data, as shown in Figure 2, which are: Single-machine and multiple-machine clustering techniques. Multiple machine clustering solutions are lately gaining attention due to their high scalability and performance response times for customers.

4.1 Single Machine Clustering Techniques

These approaches are applied to a single machine and only utilize this machine's resources. For single-machine techniques, data mining-based and dimension-reduction techniques are two common tactics [66].

1. **Data mining-based clustering**: Unsupervised classification tools are used in these strategies. Existing techniques cannot handle large volumes of detailed data since they use significant time and resources. Evolutionary-based, density-based, hierarchical-based, model-based, grid-based, and Partitioning-based procedures are all examples of data mining approaches.

2. **Partitioning-based**: this technique divides a big data set into k number of clusters (predefined by the user) by utilizing a similarity measurement which is distance, with each group representing a cluster. The

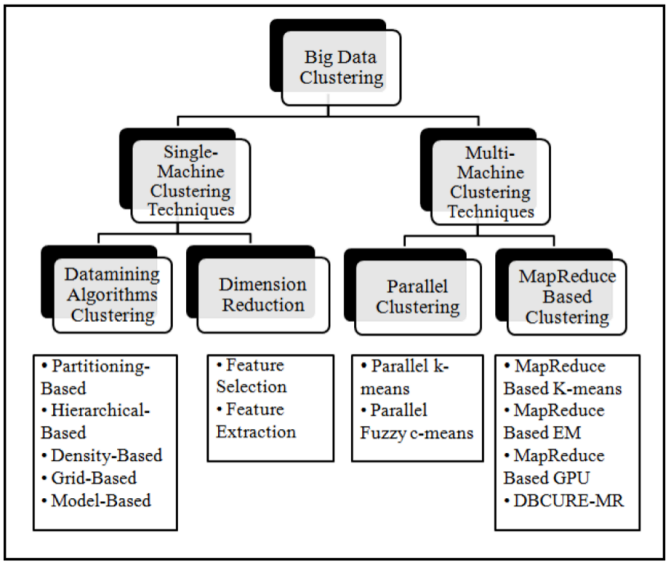

Figure 2: Big data clustering techniques

most common algorithms that use this strategy are K-Means [10], K-Medoids [64], K-Modes [71], PAM [64], CLARA [27], and CLARANS [49]. This type of clustering technique employs distance measurements as a similarity criterion, such as Manhattan distance, Euclidian distance, and maximum distance, to build clusters in a hierarchical way (tree). The two subcategories of hierarchical-based Clustering are Divisive Hierarchical and Agglomerative Hierarchical. The agglomerative hierarchical approaches begin with considering the individual point in data as a cluster, then continually fusing the most similar clusters until only one remains at the end. Two algorithms, BIRCH [54] and CHAMELEON, take advantage of this concept [16]. A divisive hierarchical clustering approach treats the entire data set as a single large cluster, then splits the most relevant clusters at each stage until a user-defined threshold of clusters is reached. This method's algorithms include PDDP [72], and DHCC [59].

3. **Density-based clustering**: Data items are organized into clusters based on density areas, connectedness, and boundary in this method. A cluster with a high density of individuals can expand in any direction. This method can detect clusters in any shape and is more resistant to noise and outliers since it is single-scanned. The typical example of this approach are: DBSCAN [45], DENCLUE [34], OPTICS [22] and DBCLASD [17].

4. **Grid-based clustering**: It uses three steps to the constructor the clusters, which are:

   – Initially, the method divides the entire area into k number of small squares, which is set by the

user and is usually considerably less than the database's size.

- Next, deleting the cells with low density of data objects;
- At the end, it merges neighboring cells with high densities to create clusters. In terms of obtaining speed, the grid-based approach provides a major advantage. On either hand, the quality of a cluster is proportional to the number of cells provided by the user at the start. The most well-known algorithms in this family are STING [20], CLIQUE [38], OPTIGrid [26], and WaveCluster [17].

5. **Model-based clustering**: these approaches provide, by allowing the preset mathematical model to adjust the data to the best of its ability, an average approximation of model parameters. To determine the classification's uncertainty, this model employs probability distributions. Automatically, the number of clusters could be found with this method, which is more outliner and noise-resistant. With the increasing number of model parameters, they become more complicated. EM [11], COBWEB [51], and MCLUST [74] are some examples of this method.

6. **Dimension reduction techniques**: directly proportional to the amount of data and the complexity and speed of this technique. The number of variables and examples or instances determines the size of a dataset. Scholars are attempting to minimize the large dimensionality of data by applying two models: feature extraction and feature selection, to make algorithms very rapid and time-effective.

   - Feature selection: This entails picking a limited number of important variables (features) from a larger pool of possibilities. The amount of chosen features in the subset can be controlled by one of the subset creation approaches in the feature choices algorithm or by the user as an input parameter. Correlation-based Feature Selection is one of three popular Sequential Forward Selection (SFS) [39], feature selection strategies (CFS) [73], and Markov Blanket Filter (MBF) [67].
   - Feature extraction's purpose is to integrate the initial set of features via operational mapping to generate new feature subsets or variables. To put it another way, the initial set of characteristics will be lowered to a new subset. Analyze the Principal Components (PCA) [43], Linear Discriminant, Singular Value Decomposition (SVD), and Analysis (LDA) [19] [70] are examples of this technique.

Figure 3 sums up the single-machine clustering techniques of big data.

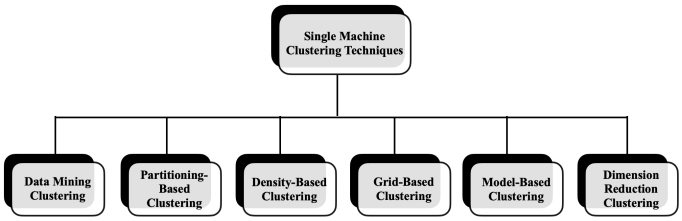

Figure 3: Single machine clustering techniques

4.2 Multiple machine clustering techniques

The massive amount of data cannot be handled by single-machine clustering algorithms (measured in Petabytes) in an adequate time with the current data explosion era. Indeed, with the advancement of computer specifications and computational sharing technologies, researchers hypothesized that similar algorithms could be executed on multiple machines by pooling their resources. These methods entail breaking down large amounts of data into smaller chunks. These components will be handled in various machines, and the machines' resources will be used to solve them. Multi-machine algorithms offer a faster processing speed and greater scalability than single-machine algorithms, but they face a significant challenge in high data traffic costs.

1. **Parallel clustering**: In data mining algorithms, there are three (3) techniques of parallelism utilized in the literature, and they are as follows [55]:

   - **Independent parallelism**: which each CPU uses the entire amount of data available, and there without intercommunication amongst processors at this level;
   - **Task parallelism**: Each processor executes a different algorithm (task).
   - **SPMD parallelism (Single Program Several Data)**: It is when the same algorithm is run on multiple processors with separate data sets.

   Parallel approaches increase Clustering's scalability and speed, but they also introduce a new responsibility for programmers: the complexity of data distribution.

   Parallel techniques increase Clustering's scalability and performance but also give programmers a new responsibility: dealing with data distribution complexity. As illustrated in Figure 4, parallel clustering operations are frequently carried out in four phases. The complete data sets are partitioned before being distributed among workstations. Each computer subsequently processes the partitioned data to form local clusters. The forward stage is to integrate the global clusters produced by previously collected local clusters. Finally, final clusters are disseminated to each computer to improve and modify the local clusters. That is a step that can be skipped.

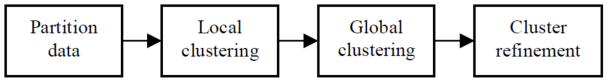

Figure 4: Parallel clustering techniques flowchart

A parallel platform is used to run several data mining clustering algorithms. Kantabutra and colleagues [62] created a new solution based on Network Of Workstations (NOWs), a parallel K-Means, in which researchers employ a message-passing mechanism for resource communication. Another approach, parallel CLARANS, was implemented in [36]. It is using a master-slave mechanism based on PVM (Parallel Virtual Machine) to make a group of a single machine or distributed computers. The message-forwarding model also establishes communication between CPUs.

[23] introduces PBIRCH, a parallel approach to the BIRCH method, in hierarchical approaches. The SPMD (Single Program Multiple Data) technique is used to achieve the parallel paradigm in this program, and the message forwarding mechanism protects communication between processors. Using a work pool analogy, a parallel variant of the CHAMELEON method was described in [16]. This approach is divided into three stages-the first concerns using the concurrent K-Nearest Neighbor approach to reduce the study's temporal complexity. Second, the previously collected sparse graph is partitioned using multilayer graph partitioning techniques to form smaller clusters. Ultimately, clusters are combined to shape more prominent clusters depending on their interconnectivity and proximity.

One of the most used parallel clustering techniques is Density-based distributed clustering (DBDC) [14], in which the entire data set is partitioned and distributed amongst sites (machines). The local model is then created by each site using the DBSCAN technique to establish a local cluster and identify representative data objects (i.e., cluster IDs). All local models are sent to a central location known as the server to generate a global model. Finally, clients get the global model on new local workstations so that local clusters may be updated and global clusters can be built.

Researchers have recently become interested in using GPUs rather than CPUs due to their high computational density per memory access. Because GPUs are designed for parallel processes, they are faster and more powerful than CPUs. The parallel GPU-based acceleration algorithm variant of DBSCAN is G-DBSCAN [53]. There are two parallelized phases in this approach. The first phase entails creating a graph whose edges are produced by adhering to a predetermined threshold. The second stage is to locate the cluster by traversing the previously generated graph using the Breadth-First Search (BFS) approach.

2. **MapReduce-based clustering**: In 2004, Google released MapReduce as an open-source system and computer program for working with massive data sets. This strategy uses the parallel paradigm to break the principal work into smaller jobs distributed across numerous processing nodes. The architecture of MapReduce is shown in Figure 5. It comprises two (2) functions with user-based scheduling (map and reduce). In each mapping machine, the first step is to analyze the scattered data. At the end of this phase, a set of pairs of intermediate keys/values is formed. By employing the reducer machines as a shuffling or combining mechanism to blend intermediate values and intermediate keys that are associated together [29]. The real benefit of this technique is that it obscures the complexities of parallel execution, allowing the user to concentrate solely on data classification and processing strategies. Furthermore, MapReduce can save programmers time by automatically managing networking issues like load balancing, data dissemination, and fault tolerance, allowing for massive parallelism and more simplicity by expanding the ability of parallel systems.

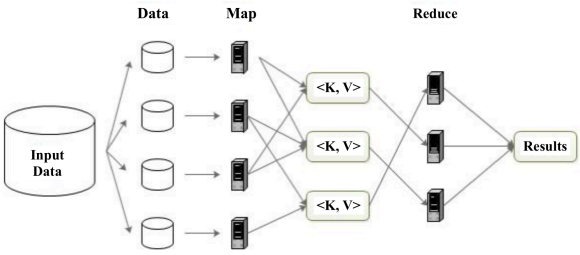

Figure 5: MapReduce framework

Various classification methods are implemented in a MapReduce platform [3], including k-means with MapReduce (PK-Means); MRDBSCAN, which is used GPU with MapReduce-based DBSCAN.

5 Big data clustering platforms

Big data platforms are divided into three categories: Batch processing, participatory analytics, and real-time handling. The most common big data platforms are Batch processing platforms, which perform numerical computations; streaming is allowed in applications that require low latency; real-time processing typically necessitates high-performance data processing, and interactive analytics allow users to remotely access datasets and perform a variety of tasks as required. The most extensively used batch-oriented platform is Apache Hadoop. Apache Hadoop is a free and

open-source MapReduce implementation. MapReduce was created by Google and is used to solve challenges such as large-scale machine learning and Clustering [52]. The map and reduce primitives in the functional language have inspired the functional language (Lisp) [57]. Hadoop breaks down an issue into the smallest unit that may be recursively handled. The little bits are subsequently spread among system nodes for execution, with a primary node, and worker nodes coordinating the operation [42]. MapReduce, HDFS, and YARN are the software components that makeup Hadoop. Abstractions like Hive, Hbase, Pig, and Spark are part of Hadoop's ecosystem. These Hadoop modules and abstractions cover the complete big data value chain, including data collection, storage, processing, evaluation, and administration. Hadoop can be deployed in huge-scale companies due to its low cost, scalability, fault tolerance, and flexibility.

Stream processing systems, often known as real-time or semi-real-time systems, are the following type of big data platform. These systems are required when data streams are continuous, and speedy processing is required. Users of real-time systems require quick access and analysis of data held in warehouses. Stream systems necessitate continuous data processing without the need to preserve it. Whether and transportation systems are two examples of such uses. The number of sensors in networked items is predicted to climb to fifty billion by 2020, these objects being linked to smart gadgets and smartphones [21]. The data from the devices might be used in both traditional and creative ways to improve the well-being of individuals, processes, the environment, systems, and organizations. These devices would need stream processing. Storm and SAP HANA are two examples of stream processing systems. Interactive analytics is the final big data platform accessible.

Academics from the University of California, Berkeley, created Spark, a next-generation prominent data processing paradigm. It is a Hadoop alternative to alleviate disk I/O constraints and boost older systems' performance. Spark's capacity to do in-memory computations is one of its most notable features. It allows data to be kept in memory for iterative operations, bypassing Hadoop's disk overhead barrier. Spark is a Java, Scala, and Python-based large-scale data processing engine that has been proven to be up to 100 times faster than Hadoop MapReduce. Data is reduced when it fits in memory; when it does not, it is up to 10 times faster. It is a Hadoop alternative designed to get around disk I/O constraints and boost the performance of older servers. Spark's ability to execute in-memory computations is one of its most distinguishing features. It enables caching data in memory, and for iterative activities, Hadoop's disk overhead barrier is bypassed. Spark is a large-scale data processing engine that supports Java, Scala, and Python and has been tested to be up to 100 times quicker than Hadoop MapReduce. When data fits in memory, it is up to ten times quicker; when data does not, it is up to ten times faster. It can read data from HDFS and execute on the Hadoop Yarn management. As a result, it may operate on a wide range of platforms.

Table 1 summing up the platforms comparison of the big data. Each platform has its advantages over the other, therefore, selecting the best platform depends on the big data characteristics and requirements [18] [13] [61] [25].

6 Clustering issues and challenges

The amount of information on the web is increasing at a breakneck speed. In the broad sense, the data can be categorized into three main categories: structured (consists entirely of basic data types such as integrals, array of integrals or characters, and characters). The unstructured (consisting of unstructured data types such as strings) is used in the SQL model. The semi-structured data consists of structured data and unstructured data types. With semi-structured, the two previous data types are combined and represented as XML in general. Most of the data created is unstructured, and standard database management solutions cannot handle it. The 3Vs defining Big Data are:

1. **Volume**: The amount of data that must be processed continuously increases. This demonstrates how big data increased use of contemporary technology (smartphones, social networks, networked equipment, and so forth) causes us to generate more and more data in our personal and professional relationships; corporations are coping with a data explosion. This Volume does keep growing at a rapid pace. The quantity of data stored worldwide is expected to increase every four years. Since 2010, it has amassed more data than it has since the start of time.

2. **Velocity**: The rate at which data is produced, gathered, and exchanged is referred to as Big Data velocity. These data are being produced and developed at a breakneck speed. As a result, real-time data gathering, analysis, and utilization should become increasingly common; it may even be possible to break data storing and instead study streaming to get the correct summing up.

3. **Variety**: The final "V" indicates that the data is not always structured and may be diverse. Indeed, it can use information from websites, posts, messages, social media exchanges (Facebook, Youtube, Whatsapp), photos, videos, audio, logs, spatial data, biometrics, and other sources. They come from various places: the web, text mining, image mining [2], and so on. To develop actionable findings, we must mix different sources. The complexity of Big Data explains why typical data warehousing infrastructure is difficult to use.

They are making heterogeneous data (climate, transportation, geography, and automobile traffic) with linking to get relevant data and enhance the different exploiting of the sectors of the large and scattered quantity of data. Indeed,

Property	Apache Spark	Hadoop MapReduce
Performance	Lightning-fast cluster computing (100 times faster)	Slower than Spark
Read/Write Cycle	Low	High
Usability	Easy	Requires code for every operation
Realtime Analysis	Supported	Not Supported
Latency	Low	High
Fault Tolerance	Supported	Supported
Security	Low	High
Cost	High (requires a lot of RAM)	Low
Developing Language	Scala	Java
Licenses	Free	Free
SQL Support	Spark SQL	Hive
Scalability	High	High
Machine Learning	MLLib	Apache Mahout
Data Caching	Supported	Not Supported
Hardware Requirements	High-Level hardware	Commodity hardware

Table 1: Big data clustering platforms

Big Data's ultimate challenge. According to the HACE theory (Heterogeneous, Autonomous, Complexity, Evolving), the big data's most significant characteristics are [56]:

1. Heterogeneous data represent the variety of data sources, including Twitter, Facebook, Instagram, and social messaging application, in a complicated and heterogeneous approach, necessitating various methodologies and solutions.

2. One of Big Data's most distinguishing features is that it is self-contained and dependent on self-contained sources. This source comprises dispersed and decentralized controls in this fashion, allowing each data source to function independently of any centralized control. Each web server on the World Wide Web (WWW) can create information and function successfully without other server assistance. On the other side, the intricacy of Big Data shows it is incredibly fragile, and it would swiftly fail if she relied on any centralized control unit. Another advantage of autonomous servers is that it allows some applications of Big Data, such as Google, and other social networks, such as Instagram, to give clients instant replies and services.

3. Data is acquired in several methods, including multitable, multi-source, multi-view, sequential, and distributed treatment data or massively parallel processing, which adds to Big Data's complexity (MapReduce). As data complexity increases with increasing Volume, traditional treatment methods, such as relational database supervisors tools, are not sufficient anymore to keep up with the requirements of storing and capturing, extra analysis, and traditional potential treatments, such as administration of database engine tools, are not sufficient anymore to hold the prerequisites of capture, backup, and extra analysis.

4. The development of complex data, which is constantly changing, is also an essential element. Big data is evolving at a breakneck pace. When a client leaves a comment on a page of a social site, over a while, these comments must be retrieved for the algorithm to function and have accurate data.

To handle the expanding demands for data, the capacity has to be enhanced and effective in practices and methodologies. To leverage the data's functions without engaging extra workers, Big Data needs innovative solutions to boost capacity and effective treatment.

Traditional data mining techniques have been unable to address critical data processing needs due to the exponential expansion of data. So, to use this massive data, effective processing, and practical computing technique is required for this complex, massive, heterogeneous, and dynamic data.

7 Recent advancements

There are many research advancements in clustering techniques due to the importance and requirements to overcome the issue in the clustering techniques which are working over big data. The most recent solution overview is presented below:

Mehdi Assefi, et al (2017).[8]. In this contribution, they have looked at the open-source framework Apache Spark MLlib 2.0 because of its scalability and platform-agnostic machine learning library from a computational standpoint. To analyze the platform's qualitative and quantitative qualities, They conduct various real-world machine learning experiments [4]. They also discuss recent trends in big data

research of machine learning and offer suggestions for futuristic studies.

Gunasekaran Manogaran, et al (2017). [48] have used a Gaussian Mixture (GM) Clustering and Bayesian hidden Markov model (HMM) technique to simulate DNA copy number variation across the genome in this work. The suggested Bayesian HMM with GM Clustering technique is compared to other approaches such as the pruned precise linear time method, binary classification method, and segment neighborhood method [15]. Experimental data support the efficiency of the suggested change detection approach.

Gourav Bathla et al. (2018). [12]. In this paper, the Kprototype algorithm is implemented using MapReduce. Experiments have shown that using Kprototype with Mapreduce improves performance on several nodes, and single nodes are compared together. For comparison, CPU running period and performance are employed as assessment measures. This work proposes an intelligent splitter that divides extensive mixed data into numerical and category data. Compared to existing methods, the proposed approach performs better when dealing with enormous amounts of data.

Ahmed Z. Skaik (2018) [65] has presented a new approach that overcomes the shortcomings of both algorithms, improves significant data clustering, and avoids being trapped in a locally optimal solution by leveraging a robust optimization algorithm (Particle Swarm Optimization) known as PSO and reducing time and resource consumption by leveraging a robust distribution framework Apache/Spark. Research findings show that the algorithm can significantly lower clustering costs and create superior clustering outputs more accurately and fruitfully than solo K-Means, IWC, and PSO methods.

Behrooz Hosseini, et al (2018). [32] proposed a solution built and tested using the Apache Spark framework with a range of datasets. Each phase of the proposed technique for all points is independent, and there are no serial bottlenecks in the clustering operation. Other data points are members of the cluster with the closest center, filtering out outliers and improving the robustness of the recommended technique. The proposed technique has been tested and compared to other recently published studies. The cluster validity indices of the proposed technique reveal its advantage in accuracy and noise resilience compared to previous research. Compared to others techniques, the suggested method outperforms them in terms of scalability, performance, and computing time.

Mo Haia, et al, (2018). [28]. The performance of the main large data processing platforms: Hadoop and Spark, are examined using parallel clustering: parallel K-means with and without fuzzy. Experiments are conducted on a wide range of text and numerical collections and clusters of varying sizes. The findings reveal that: (1) with a 6 GB of RAM in each node, they achieve a 60% performance gain over Hadoop and a 32% performance improvement over Spark for the same data set; (2) the 6 GB of RAM should be selected over 3 GB of RAM for high clustering performances.

Aditya Sarma, et al, (2019). [63] offer DBSCAND, a highly scalable distributed implementation of DBSCAN that takes advantage of commodity cluster hardware, in this work. Experimental results show that the suggested algorithms perform significantly better than the respective state-of-the-art techniques for various typical datasets. DBSCAN-D is a DBSCAN exact parallel solution that can quickly analyze enormous volumes of data (1 billion data points in 41 minutes on a 32-node cluster) while maintaining the same Clustering as regular DBSCAN.

Omkaresh Kulkarni, et al, (2020). [44]. To lower computational complexity, the clustering approach is critical in this study. With the understanding of clustering algorithms, the MapReduce architecture is used to process huge data from distributed sources (MRF). One function of the MRF is for mapping, while the other is for reducing, the ideal centroids are computed in the mapper phase using the proposed approach, which is then improved in the reducer phase. In the experiment, the Skin data set and the geolocation data set from the UCI machine learning repository were employed, and accuracy and the DB Index were used to develop the inquiry. The proposed method analyses acquired a maximum accuracy with proven up to 90.6012% with a minimal DB Index of 5.33.

Hoill Jung, et al, (2020). [40] have presented a method for establishing trustworthy user modeling in this study by integrating standard static model information with information acquired from social networks, and a different amount of weight is applied based on the users' associations. PrefixSpan is employed in a life care forecasting model that uses social relations to supplement a weak area in the candidate pattern that frequently takes a long time to scan. They compared the common cluster technique with a social mining-based approach for performance measurement. As a result, the suggested technique in the mining-based healthcare platform outperformed the existing model-based cluster method.

K. Rajendra Prasad, et al, (2021). [56], the suggested technique in this study solves the big clustering problem by deriving sampling-based crisp partitions. The crisp partitions will reliably predict the cluster labels of data items. This study uses large real-world synthetic datasets to demonstrate the suggested work's performance efficiency.

Mustafa Razee et al. (2021) [58], as a novel way, suggested the game-based k-means (GBK-means) algorithm. To demonstrate the superiority and efficiency of GBK-means over conventional clustering algorithms, namely k-means and fuzzy k-means, They use the following syntactic and real-world data sets: (1) a series of two-dimensional syntactic data sets; and (2) ten benchmark data sets that are widely used in different clustering studies. GBK-means can cluster data more accurately than classical algorithms based on eight evaluation metrics, including the F-measure, the Dunn index (DI), the rand index (RI), the Jaccard index (JI), normalized mutual information (NMI), normalized varia-

tion of information (NVI), a measure of concordance, and error rate (ER).

Chen Zhen (2021) [77] this research proposes a big data fuzzy K-means clustering and information fusion-based English teaching competence evaluation system. The results demonstrate that adopting this technique to evaluate English teaching ability enhances learning ability accuracy and resource efficiency.

C. Wu. et al. (2021) [69] have set up random sampling before parallelizing the distance computing technique, ensuring data item independence and parallel cluster analysis. After the MapReduce parallel processing, They use numerous nodes to compute distance, which enhances the algorithm's speed. Finally, the grouping of data objects is parallelized. The findings indicate that the system can provide timely, consistent, and convergent services.

Fouad H. et al. (2022) [9] have proposed a solution to run the clustering technique on a single-instruction machine processor found in the mobile device's processor. Big data clustering across the network may be efficiently handled using k-means Clustering in a distributed method based on mobile machine learning. The results revealed that using a neural engine processor on a mobile smartphone or tablet can increase the speed of the clustering algorithm by up to two times compared to traditional laptop/desktop processors, resulting in a performance improvement of the cluttering algorithm of up to two times faster. Furthermore, compared to parallel and distributed k-means, the number of iterations necessary to create (k) clusters was reduced by up to two-fold.

Lin Ma et al. (2022) [47] start by using a quick mean-shift technique with configurable radius and active subsets to find the centers, which significantly cuts down on processing time. The second stage employs the form of the probability density function of the distribution of distances between a selected point and the other points in the data set. That is done to determine the critical and cluster radiuses of the fast mean-shift method. The new algorithm has four distinct benefits. It reduces processing complexity, overcomes dimensionality difficulties, can handle a wide range of spherical data sets, and is noise and outlier tolerant. They discovered that the proposed technique works effectively after testing it on several synthetic and real-world data sets.

The results of this literature review are summarized in Tables 2 and 3.

References	Clustering Technique	Goal	Results
Mehdi Assefi, et al (2017) [8].	MLib 2.0 Apache Spark	Evaluating the platform qualitative and quantitative attributes	Addressing the recent solutions in big data machine learning
Gunasekaran Manogaran, et al (2017) [48].	Bayesian Hidden Markov Model (HMM) with Gaussian Mixture (GM) Clustering	Modeling the DNA copy number change across the genome	Experimental results demonstrate the effectiveness of the proposed change detection algorithm.
Gourav Bathla, et al.(2018) [12].	K-Prototype with Map-Reduce	splits mixed big data into numerical and categorical data	Proving that the proposed algorithm works better for large-scale data.
Ahmed Z. Skaik (2018) [65]	IWC and PSO	Introduce a new approach that conquers the drawbacks of both algorithms	Reduces the clustering cost, and produces superior clustering outputs
Behrooz Hosseini, et al (2018). [32]	Apache Spark framework and Locality Sensitive Hashing	it has been used in gene expression clustering as a sample of its application	shows the proposed method's superior scalability, high performance, and low computation cost.
Mo Haia, et al, (2018). [28]	Parallel Clustering Technique (K-Mean and Fuzzy K-Mean)	to obtain a high clustering performance	60% performance improvement compared with Hadoop, and achieve about 32% performance improvement compared with Spark
Aditya Sarma, et al, (2019) [63].	DBSCAN	to exploit a commodity cluster infrastructure	ability of massive data processing efficiently (1 billion data points in 41 minutes on a 32 node cluster).
Omkaresh Kulkarni, et al, (2020) [44].	Fractional Sparse Fuzzy C-Means and PSO	Relieving the computational complexity of the clustering method	Acquired a maximum accuracy of 90.6012% and a minimum DB Index of 5.33.
Hoill Jung et al., (2020) [40]	Conventional Static Model	to create reliable user modeling and apply a different weight depending on user relations.	mining-based healthcare platform had better performance than the conventional model-based cluster method [6].

Table 2: Big data clustering recent advancements

References	Clustering Technique	Goal	Results
K. Rajendra Prasad et al., (2021) [56]	BigVAT with the derivation of sampling-based crisp partitions	The crisp partitions will accurately predict the cluster labels of data objects.	addresses the clustering problem of bigVAT
Mustafa Razee et al. (2021) [58]	Game-Based K-Means (GBK-means)	Improving K-means Performance	GBK-means can cluster data more accurately than classical algorithms based on eight evaluation metrics
Chen Zhen, (2021). [77].	clustering fuzzy K-means technique	an English teaching ability evaluation	improves the accuracy and the efficiency of learning techniques.
C Wu. el. al. (2021) [69]	K-Means Clustering Based on Distributed Computing	Improving k-Means clustering performance	Improving the performance and stably of the parallelized k-means clustering
Fouad H. et. al. (2022) [9]	Neural-Engine Based k-means clustering	performance improvement	improving the performance of clustering up to 200%
Lin Ma et al. (2022) [47]	Radar Scanning Strategy Clustering	Reducing computational power and overcomes the issues of the high dimensionality of radar scanning technique	Improves the accuracy, efficiency, and performance of radar scanning technique.

Table 3: Big data clustering recent advancements

8 Conclusion

Big data has become an essential part of research and engineering in today's world. Organizations are increasingly relying on these massive data sets to gain new insights and explore new possibilities. However, traditional machine learning approaches are often insufficient in dealing with the massive volume, velocity, variety, veracity, and value of big data, commonly referred to as the 5 Vs of Big Data. Consequently, machine learning algorithms need to reimagine themselves to meet these challenges. The integration of machine learning and big data has paved the way for a promising future in the data-driven industry. Big data processing and machine learning algorithms work hand in hand to reveal new information, increase productivity, and generate new and unexpected insights. Clustering techniques such as K-means, Fuzzy C-means, and K-mode are widely used in numerous frameworks for big data clustering. Python has emerged as a valuable tool for executing big data clustering techniques. Researchers rely on Python for its simplicity, flexibility, and scalability, making it an excellent option for big data processing. However, alternative frameworks for big data techniques are also necessary, given the ever-increasing complexity of big data sets. Despite the growing importance of big data clustering, there is a limited number of essential publications on the topic. One of the main reasons for this is the difficulty of obtaining research data that are suitable for implementing big data techniques. Nonetheless, previous studies have made significant contributions to evaluating the performance, efficiency, and accuracy of big data clustering and reducing noise and processing time. In conclusion, big data clustering is a critical aspect of the data-driven industry. As the amount of data continues to grow, it is essential to continually refine machine learning algorithms and explore alternative frameworks to meet the challenges posed by big data. The use of Python and other tools will be crucial in developing efficient and effective big data clustering techniques.

References

[1] Omran Alshamma, Fouad H Awad, Laith Alzubaidi, Mohammed A Fadhel, Zinah Mohsin Arkah, and Laith Farhan. Employment of multi-classifier and multi-domain features for pcg recognition. In *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, pages 321–325. IEEE, 2019. doi:10.1109/DeSE.2019.00066.

[2] Laith Alzubaidi, Muthana Al-Amidie, Ahmed Al-Asadi, Amjad J Humaidi, Omran Al-Shamma, Mohammed A Fadhel, Jinglan Zhang, J Santamara, and Ye Duan. Novel transfer learning approach for medical imaging with limited labeled data. *Cancers*, 13(7).

[3] Laith Alzubaidi, Mohammed A Fadhel, Omran Al-Shamma, Jinglan Zhang, and Ye Duan. Deep learning models for classification of red blood cells in microscopy images to aid in sickle cell anemia diagnosis. *Electronics*, 9(3):427, 2020. doi:10.3390/electronics9030427.

[4] Laith Alzubaidi, Mohammed A Fadhel, Omran Al-Shamma, Jinglan Zhang, J Santamaría, and Ye Duan. Robust application of new deep learning tools: an experimental study in medical imaging. *Multimedia Tools and Applications*, pages 1–29, 2021. doi:10.1007/s11042-021-10942-9.

[5] Laith Alzubaidi, Reem Ibrahim Hasan, Fouad H Awad, Mohammed A Fadhel, Omran Alshamma, and Jinglan Zhang. Multi-class breast cancer classification by a novel two-branch deep convolutional neural network architecture. In *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, pages 268–273. IEEE, 2019. doi:10.1109/DeSE.2019.00057.

[6] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1):1–74, 2021. doi:10.1186/s40537-021-00444-8.

[7] Hiba Asri, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. Big data in healthcare: challenges and opportunities. pages 1–7, 2015. doi:10.1109/CloudTech.2015.7337020.

[8] Mehdi Assefi, Ehsun Behravesh, Guangchi Liu, and Ahmad P Tafti. Big data machine learning using apache spark mllib. In *2017 ieee international conference on big data (big data)*, pages 3492–3498. IEEE, 2017. doi:10.1109/BigData.2017.8258338.

[9] Fouad H Awad and Murtadha M Hamad. Improved k-means clustering algorithm for big data based on distributed smartphone neural engine processor. *Electronics*, 11(6):883, 2022. doi:10.3390/electronics11060883.

[10] Fouad H Awad, Murtadha M Hamad, and Laith Alzubaidi. Robust classification and detection of big medical data using advanced parallel k-means clustering, yolov4, and logistic regression. *Life*, 13(3):691, 2023.

[11] S. Balakrishnan, M.J. Wainwright, and B. Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120,. doi:10.1214/16-AOS1435.

[12] Gourav Bathla, Himanshu Aggarwal, and Rinkle Rani. A novel approach for clustering big data based on mapreduce. *International Journal of Electrical*

& Computer Engineering (2088-8708), 8(3), 2018. doi:10.11591/ijece.v8i3.pp1711-1719.

[13] Yassine Benlachmi and Moulay Lahcen Hasnaoui. Big data and spark: Comparison with hadoop. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 811–817. IEEE, 2020. doi:10.1109/WorldS450073.2020.9210353.

[14] Panthadeep Bhattacharjee and Pinaki Mitra. A survey of density based clustering algorithms. *Frontiers of Computer Science*, 15(1):1–27, 2021. doi:10.1007/s11704-019-9059-3.

[15] Salima Bourougaa-Tria, Farid Mokhati, HoussemEddine Tria, and Okba Bouziane. Spubbin: Smart public bin based on deep learning waste classification an iot system for smart environment in algeria. *Informatica*, 46(8), 2022. doi:10.31449/inf.v46i7.4331.

[16] X. Cao, T. Su, P. Wang, G. Wang, Z. Lv, and X. Li. An optimized chameleon algorithm based on local features. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 184–192. ACM. doi:10.1145/3195106.3195118.

[17] Matthias Carnein and Heike Trautmann. Optimizing data stream representation: An extensive survey on stream clustering algorithms. *Business & Information Systems Engineering*, 61(3):277–297, 2019. doi:10.1007/s12599-019-00576-5.

[18] Eduardo PS Castro, Thiago D Maia, Marluce R Pereira, Ahmed AA Esmin, and Denilson A Pereira. Review and comparison of apriori algorithm implementations on hadoop-mapreduce and spark. *The Knowledge Engineering Review*, 33, 2018. doi:10.1017/S0269888918000127.

[19] D. Chu, L.-Z. Liao, M.K.-P. Ng, and X. Wang. Incremental linear discriminant analysis: a fast algorithm and comparisons. *IEEE transactions on neural networks and learning systems*, 26(11):2716–2735,. doi:10.1109/TPAMI.2005.244.

[20] Nguyen Duy Dat, Vo Ngoc Phu, Vo Thi Ngoc Tran, Vo Thi Ngoc Chau, and Tuan A Nguyen. Sting algorithm used english sentiment classification in a parallel environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(07):1750021, 2017. doi:10.1142/S0218001417500215.

[21] Marcos Dias de Assuncao, Alexandre da Silva Veith, and Rajkumar Buyya. Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. *Journal of Network and Computer Applications*, 103:1–17, 2018. doi:10.1016/j.jnca.2017.12.001.

[22] Z. Deng, Y. Hu, M. Zhu, X. Huang, and B. Du. A scalable and fast optics for clustering trajectory big data. *Cluster Computing*, 18(2):549–562,. doi:10.1007/s10586-014-0413-9.

[23] Kheyreddine Djouzi and Kadda Beghdad-Bey. A review of clustering algorithms for big data. In *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–6. IEEE, 2019. doi:10.1109/ICNAS.2019.8807822.

[24] Jason Ernst, Gerard J Nau, and Ziv Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21(suppl 1):i159–i168, 2005. doi:10.1093/bioinformatics/bti1022.

[25] Mithu Mary George and PS Rasmi. Performance comparison of apache hadoop and apache spark for covid-19 data sets. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 1659–1665. IEEE, 2022. doi:10.1109/icssit53264.2022.9716232.

[26] Attri Ghosal, Arunima Nandy, Amit Kumar Das, Saptarsi Goswami, and Mrityunjoy Panday. A short review on different clustering techniques and their applications. *Emerging technology in modelling and graphics*, pages 69–83, 2020. doi:10.1007/978-981-13-7403-6_9.

[27] T. Gupta and S.P. Panda. A comparison of k-means clustering algorithm and clara clustering algorithm on iris dataset. *International Journal of Engineering & Technology*, 7(4):4766–4768,. doi:10.14419/ijet.v7i4.21472.

[28] Mo Hai, Yuejing Zhang, and Haifeng Li. A performance comparison of big data processing platform based on parallel clustering algorithms. *Procedia computer science*, 139:127–135, 2018. doi:10.1016/j.procs.2018.10.228.

[29] Murtadha M Hamad. A comparative study of indexing techniques effect in big data system storage optimization. In *2020 2nd Al-Noor International Conference for Science and Technology (NICST)*, pages 18–21. IEEE, 2020. doi:10.1109/NICST50904.2020.9280309.

[30] Richard J Hathaway and James C Bezdek. Extending fuzzy and probabilistic clustering to very large data sets. *Computational Statistics and Data Analysis*, 51(1):215–234, 2006. doi:10.1016/j.csda.2006.02.008.

[31] Timothy C Havens, James C Bezdek, and Marimuthu Palaniswami. Scalable single linkage hierarchical clustering for big data. In *2013 IEEE eighth international conference on intelligent sensors, sensor networks and information processing*, pages 396–401. IEEE, 2013. doi:10.1109/issnip.2013.6529823.

[32] Behrooz Hosseini and Kourosh Kiani. A robust distributed big data clustering-based on adaptive density partitioning using apache spark, 2018. doi:10.3390/sym10080342.

[33] Hassan Ibrahim Hayatu, Abdullahi Mohammed, and Ahmad Barroon Isma'eel. Big data clustering techniques: Recent advances and survey. *Machine Learning and Data Mining for Emerging Trend in Cyber Dynamics*, pages 57–79, 2021. doi:10.1007/978-3-030-66288-2_3.

[34] A. Idrissi, H. Rehioui, A. Laghrissi, and S. Retal. An improvement of denclue algorithm for the data clustering. In *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA*, pages 1–6. IEEE. doi:10.1109/ICTA.2015.7426936.

[35] Flix Iglesias and Wolfgang Kastner. Analysis of similarity measures in times series clustering for the discovery of building energy patterns. *Energies*, 6(2):579–597, 2013. doi:10.3390/en6020579.

[36] K Indira, S Karthiga, CV Nisha Angeline, and C Santhiya. Parallel clarans algorithm for recommendation system in multi-cloud environment. In *Computer Networks and Inventive Communication Technologies*, pages 461–472. Springer, 2021. doi:10.1007/978-981-15-9647-6_36.

[37] Manaswini Jena, Debahuti Mishra, Smita Prava Mishra, Pradeep Kumar Mallick, and Sachin Kumar. Exploring the parametric impact on a deep learning model and proposal of a 2-branch cnn for diabetic retinopathy classification with case study in iot-blockchain based smart healthcare system. *Informatica*, 46(2), 2022. doi:10.31449/inf.v46i2.3906.

[38] Yan Jin, Bowen Xiong, Kun He, Yangming Zhou, and Yi Zhou. On fast enumeration of maximal cliques in large graphs. *Expert Systems with Applications*, 187:115915, 2022. doi:10.1016/j.eswa.2021.115915.

[39] A. Jovic, K. Brkic, and N. Bogunovic. A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO*, pages 1200– 1205. IEEE. doi:10.1109/MIPRO.2015.7160458.

[40] Hoill Jung. Social mining-based clustering process for big-data integration. *Journal of Ambient Intelligence and Humanized Computing*. doi:10.1007/s12652-020-02042-7.

[41] Harihar Kalia, Satchidananda Dehuri, and Ashish Ghosh. A survey on fuzzy association rule mining. *International Journal of Data Warehousing and Mining (IJDWM)*, 9(1):1–27, 2013. doi:10.4018/jdwm.2013010101.

[42] Taiwo Kolajo, Olawande Daramola, and Ayodele Adebiyi. Big data stream analysis: a systematic literature review. *Journal of Big Data*, 6(1):1–30, 2019. doi:10.1186/s40537-019-0210-7.

[43] X. Kong, C. Hu, and Z. Duan. Generalized principal component analysis. In *Principal Component Analysis Networks and Algorithms*, pages 185–233. Springer. doi:10.1007/978-981-10-2915-8_7.

[44] Omkaresh Kulkarni. Mapreduce framework based big data clustering using fractional integrated sparse fuzzy c means algorithm. *IET Image Process*, 14(12):2719–2727. doi:10.1049/iet-ipr.2019.0899.

[45] K.M. Kumar and A.R.M. Reddy. A fast dbscan clustering algorithm by accelerating neighbor searching using groups method. *Pattern Recognition*, 58:39–48,. doi:10.1016/j.patcog.2016.03.008.

[46] In Lee. Big data: Dimensions, evolution, impacts, and challenges. *Business horizons*, 60(3):293–303, 2017. doi:10.1016/j.bushor.2017.01.004.

[47] Lin Ma, Yi Zhang, Víctor Leiva, Shuangzhe Liu, and Tiefeng Ma. A new clustering algorithm based on a radar scanning strategy with applications to machine learning data. *Expert Systems with Applications*, 191:116143, 2022. doi:10.1016/j.eswa.2021.116143.

[48] Gunasekaran Manogaran, V Vijayakumar, R Varatharajan, Priyan Malarvizhi Kumar, Revathi Sundarasekar, and Ching-Hsien Hsu. Machine learning based big data processing framework for cancer diagnosis using hidden markov model and gm clustering. *Wireless personal communications*, 102(3):2099–2116, 2018. doi:10.1007/s11277-017-5044-z.

[49] L. Matioli, S. Santos, M. Kleina, and E. Leite. A new algorithm for clustering based on kernel density estimation. *Journal of Applied Statistics*, 45(2):347–366,. doi:10.1080/02664763.2016.1277191.

[50] Francois G Meyer and Jatuporn Chinrungrueng. Spatiotemporal clustering of fmri time series in the spectral domain. *Medical Image Analysis*, 9(1):51–68, 2005. doi:10.1016/j.media.2004.07.002.

[51] N. Mulani, A. Pawar, P. Mulay, and A. Dani. Variant of cobweb clustering for privacy preservation in cloud db querying. *Procedia Computer Science*, 50:363–368,. doi:10.1016/j.procs.2015.04.034.

[52] Ruba Obiedat and Sara Amjad Toubasi. A combined approach for predicting employees' productivity based on ensemble machine learning methods. *Informatica*, 46(5), 2022. doi:10.31449/inf.v46i5.3839.

[53] Madhav Poudel and Michael Gowanlock. Cuda-dclust+: Revisiting early gpu-accelerated dbscan clustering designs. In *2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, pages 354–363. IEEE, 2021. doi:10.1109/HiPC53243.2021.00049.

[54] Vishnu Priya and A Vadivel. User behaviour pattern mining from weblog. *International Journal of Data Warehousing and Mining (IJDWM)*, 8(2):1–22, 2012. doi:10.4018/jdwm.2012040101.

[55] T Ragunthar, P Ashok, N Gopinath, and M Subashini. A strong reinforcement parallel implementation of k-means algorithm using message passing interface. *Materials Today: Proceedings*, 46:3799–3802, 2021. doi:10.1016/j.matpr.2021.02.032.

[56] K Rajendra Prasad, Moulana Mohammed, LV Narasimha Prasad, and Dinesh Kumar Anguraj. An efficient sampling-based visualization technique for big data clustering with crisp partitions. *Distributed and Parallel Databases*, 39(3):813–832, 2021. doi:10.1007/s10619-021-07324-3.

[57] Sreekanth Rallapalli, RRb Gondkar, and Uma Pavan Kumar Ketavarapu. Impact of processing and analyzing healthcare big data on cloud computing environment by implementing hadoop cluster. *Procedia Computer Science*, 85:16–22, 2016. doi:10.1016/j.procs.2016.05.171.

[58] Mustafa Jahangoshai Rezaee, Milad Eshkevari, Morteza Saberi, and Omar Hussain. Gbk-means clustering algorithm: An improvement to the k-means algorithm based on the bargaining game. *Knowledge-Based Systems*, 213:106672, 2021. doi:10.1016/j.knosys.2020.106672.

[59] Maurice Roux. A comparative study of divisive and agglomerative hierarchical clustering algorithms. *Journal of Classification*, 35(2):345–366, 2018. doi:10.48550/arXiv.1506.08977.

[60] Mozamel M Saeed, Zaher Al Aghbari, and Mohammed Alsharidah. Big data clustering techniques based on spark: a literature review. *PeerJ Computer Science*, 6:e321, 2020. doi:10.7717/peerj-cs.321.

[61] Yassir Samadi, Mostapha Zbakh, and Claude Tadonki. Performance comparison between hadoop and spark frameworks using hibench benchmarks. *Concurrency and Computation: Practice and Experience*, 30(12):e4367, 2018. doi:10.1002/cpe.4367.

[62] Tanvir Habib Sardar and Zahid Ansari. An analysis of mapreduce efficiency in document clustering using parallel k-means algorithm. *Future Computing and Informatics Journal*, 3(2):200–209, 2018. doi:10.1016/j.fcij.2018.03.003.

[63] Aditya Sarma, Poonam Goyal, Sonal Kumari, Anand Wani, Jagat Sesh Challa, Saiyedul Islam, and Navneet Goyal. μdbscan: an exact scalable dbscan algorithm for big data exploiting spatial locality. In *2019 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 1–11. IEEE, 2019. doi:10.1109/cluster.2019.8891020.

[64] Erich Schubert and Peter J Rousseeuw. Faster k-medoids clustering: improving the pam, clara, and clarans algorithms. In *International conference on similarity search and applications*, pages 171–187. Springer, 2019. doi:10.1007/978-3-030-32047-8_16.

[65] Ahmed Z. Skaik. Clustering big data based on iwc-pso and mapreduce. In *Thesis Submitted in Partial Fulfillment of the Requirements For the Degree of Master in Computer Engineering*. doi:10.11591/ijece.v8i3.pp1711-1719.

[66] Jing Wang, Roobaea Alroobaea, Abdullah M Baqasah, Anas Althobaiti, and Lavish Kansal. Study on library management system based on data mining and clustering algorithm. *Informatica*, 46(9), 2023. doi:10.31449/inf.v46i9.3858.

[67] Y. Wang, J. Wang, H. Liao, and H. Chen. An efficient semisupervised representatives feature selection algorithm based on information theory. *Pattern Recognition*, 61:511–523,. doi:10.1016/j.patcog.2016.08.011.

[68] Philicity K Williams, Caio V Soares, and Juan E Gilbert. A clustering rule based approach for classification problems. *International Journal of Data Warehousing and Mining (IJDWM)*, 8(1):1–23, 2012. doi:10.4018/jdwm.2012010101.

[69] Chunqiong Wu, Bingwen Yan, Rongrui Yu, Baoqin Yu, Xiukao Zhou, Yanliang Yu, and Na Chen. k-means clustering algorithm and its simulation based on distributed computing platform. *Complexity*, 2021, 2021. doi:10.1155/2021/9446653.

[70] T. Wu, S.A.N. Sarmadi, V. Venkatasubramanian, A. Pothen, and A. Kalyanaraman. Fast svd computations for synchrophasor algorithms. *IEEE Transactions on Power Systems*, 31(2):1651–1652,. doi:10.1109/TPWRS.2015.2412679.

[71] Fanyi Xie. Semiconductor scheduling problem based on k-mode clustering algorithm. In *International Conference on Frontier Computing*, pages 867–873, 2020.

[72] T. Xiong, S. Wang, A. Mayers, and E. Monga. Dhcc: Divisive hierarchical clustering of categorical data. *Data Mining and Knowledge Discovery*, 24(1):103–135,. doi:10.1007/s10618-011-0221-2.

[73] Q. Zhang, C. Zhu, L.T. Yang, Z. Chen, L. Zhao, and P. Li. An incremental cfs algorithm for clustering large data in industrial internet of things. *IEEE Transactions on Industrial Informatics*, 13(3):1193–1201,. doi:10.1109/TII.2017.2684807.

[74] Wanli Zhang and Yanming Di. Model-based clustering with measurement or estimation errors. *Genes*, 11(2):185, 2020. doi:10.3390/genes11020185.

[75] Yonglai Zhang and Yaojian Zhou. Review of clustering algorithms. *Journal of Computer Applications*, 39(7):1869, 2019. doi:10.11772/j.issn.1001-9081.2019010174.

[76] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine learning*, 55(3):311–331, 2004. doi:10.1023/B:MACH.0000027785.44527.d6.

[77] Chen Zhen. Using big data fuzzy k-means clustering and information fusion algorithm in english teaching ability evaluation. *Complexity*, 2021, 2021. doi:10.1155/2021/5554444.