International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 05 | May 2023 www.irjet.net p-ISSN: 2395-0072

Survey on MapReduce in Big Data Clustering using Machine Learning Algorithms

[1]S.Jayabharathi, [2]Dr.M.Logambal,

[1]Research Scholar, Department of Computer Science with Data Analytics, Vellalar College for Women, Thindal, Erode, Tamil Nadu, India.
[2]Assistant Professor, Department of Computer Science with Data Analytics, Vellalar College for Women, Thindal, Erode, Tamil Nadu, India.

Abstract: Higher dimensional information is characterized by enormous dimensionality of structure, spreads a high degree of difficulty, and must be understood in all these times. As the dimensionality of the dataset increases, the model data representation becomes sparse and the domain density increases, which becomes an additional task. However, when dealing with high dimensional data, it is not possible to achieve good results. However, the dimensional subspace falloff leads to a very difficult problem as well. This broadside offers limited knowledge of effective clustering. Big data analysis and processing requires a lot of effort, tools, and equipment. Hadoop, Apache, and Spark framework software use MapReduce models to perform large-scale data analysis through parallel processing and retrieve results as fast as possible. However, in the era of big data, traditional data analysis methods may not be able to manage and process large amounts of data. In order to develop efficient processing of big data, this paper improves the use of map-reduce techniques for processing big data in machine learning algorithms.

Keywords: Big Data, Map Reduce, clustering, machine learning, KNN,SVM,K-Means, Naïve Bayes, FCM

I.INTRODUCTION

Big data is an important concept for industry, academics, and researchers in the fields of computing, economics, and software development. It represents the raw materials used for processing and analysis, and the information obtained represents the results after these processing. The context of big data concerns his 7Vs (Quantity, Value, Velocity, Variety, Validity, Truth, and Visualization) of digital data generated and collected from various sources [1]. The question turns to the question of what to do with these huge amounts of data. Scientists and researchers consider big data to be one of the most important topics in computer science today. Social networking sites such as Facebook and Twitter have billions of users and generate hundreds of gigabytes of content every minute. Retailers continuously collect data from their customers. On YouTube he has 1 billion unique his users generating 100 hours of video every hour. The Content ID service is scanning your video. Over 400 years every day [2][3]. Dealing with this avalanche of data requires powerful knowledge discovery tools. Data mining techniques are known knowledge discovery tools for this purpose [4]. One of them, clustering, is defined as a method of dividing data into groups so that objects within each group are more similar than other objects within other groups. Data clustering is a well-known technique in many areas of computer science and related fields. Data mining can be seen as the main origin of clustering, but it is widely used in other research fields such as bioinformatics, energy research, machine learning, networks and pattern recognition, so much research has been done in this field. [5]. From the beginning, researchers explored clustering algorithms to manage complexity and computational load, resulting in increased scalability and speed.

The concept of machine learning is not new in the field of computing, but it has emerged as an entirely new "avatar" for the ever-changing demands of today's world. Everyone is now talking about ML-based solution strategies for a particular problem. ML is a subset of artificial intelligence that uses computer algorithms to learn autonomously from data and information. With the advent of the Internet, a lot of digital information has been created. This means more data for machines to analyze and “learn” [6]. As a result, we are seeing a resurgence in machine learning. Today, machine learning algorithms enable computers to communicate with humans, drive self-driving cars, write and publish sports game reports, and spot terrorist suspects. Machine learning (ML) is the fastest growing field in computer science [7]. Classification [8], regression [9], topic modeling [10], time series analysis, cluster analysis, association rules, collaborative filtering, and dimensionality reduction are some of the common machine learning techniques/methods. [11][12][13]14]15]. Big data is a large collection of data sets that are complex to process. Organizations struggle with creating, manipulating, and managing large datasets [16]. This data can be analyzed using software tools as part of advanced analytics capabilities such as predictive analytics, data mining, text analytics, and statistical analytics. Examples of such large amounts of data are

Facebook, Google, etc. A collection of data that may not fit well into traditional databases in semi-structured and unstructured formats. For storing unstructured data, Hadoop, Apache, and Spark are built to store and compute data in a parallel distributed environment. One such option is to distribute the work in parallel over many computers. Clustering is an important task in data mining. It is a popular data analysis technique in many fields such as information retrieval, image analysis, machine learning, and bioinformatics. Many of the efficient machine learning algorithms discussed in this article. K means Algorithm, KNN, Naive Bayes, SVM, Random Forest, Fuzzy C means Algorithm. A Map-Reduce clustering algorithm that attempts to partition a given, unspecified dataset into a fixed number (k) of clusters, suitable for parallelizing the Map-Reduce algorithms described in this article.

II. LITERATURE REVIEW

Yongyi Li etal.,[17] Research proposes a parallel k-means algorithm based on MapReduce big data clustering. First, the partition, communication, combination, and mapping models are created according to the properties of the MapReduce framework. Then, a parallel k-means algorithm based on MapReduce big data clustering was designed, and the algorithm execution process was analyzed. Finally, it is demonstrated through data and experimental analysis that the MR-K-means parallel algorithm reduces the temporal and spatial complexity and the rate of missing data points compared to traditional algorithms. According to the characteristics of big data clustering, we analyze the feasibility of parallel algorithms based on MapReduce K-means and discuss the technical strategies of parallel algorithms. Based on traditional K-Means algorithms, models for splitting, communicating, joining, and mapping are built, and parallel K-Means algorithm steps based on MapReduce are designed. The operation process and experimental results show that the K-Means parallel algorithm based on the MapReduce framework has more advantages in dealing with the ocean data clustering problem in terms of time complexity compared with the K-Means algorithm. indicates that there is Setting an appropriate number of clusters, data block size, number of iterations and number of nodes can reduce the computational and data loading space of traditional k-means algorithm, improve the efficiency of clustering, and reduce the time, space complexity. , and the rate of lost data points. .

Lei Chen etal.,[18] A particular new partitioner based on the Naive Bayes classifier, namely BAPM, achieves better performance by leveraging the Naive Bayes classifier to optimize data locality and data bias. H. Consider order type and bandwidth as classification attributes. Reduce-side locality and data skew are two important factors that affect MapReduce. Experiments have shown that the processing order of the two factors affects the execution time at different bandwidths. In this work, we develop a bandwidth-aware partitioner using a simple Bayesian classifier by considering bandwidth and job type as classification attributes for choosing an algorithm (LRS or SRL) appropriately between different bandwidths. , that is, BAPM was proposed.

F.ouatik et al.,[19] Impact on student learning and research requires extremely powerful tools to analyze this vast amount of data. So, for this study, he classified students into four classes: science, literature, technology, and creativity. There are many different types of classification. This research, which focuses on student classification, is based on classification algorithms and big data tools that can analyze large datasets using Hadoop Distributed File System 'HDFS'41. manage and analyze. and her MapReduce model for parallelism to evaluate the performance of the classification algorithm.

This shows the power of the Naive Bayes algorithm. Comparing Naive Bayes, Neural Networks, and K-Nearest Neighbors in run time and accuracy reveals that Bayes is best when characterized by speed, according to processed data and deployed device characteristics Did. This is necessary to make quality decisions in a short amount of time.

Xuesong Yan et al.,[20] We designed a parallel MapReduce-based ANN join algorithm for big data classification. In our research, we further implemented the algorithm on a cluster of nine virtual machines using Hadoop. Experimental results show that MapReduce-based ANN joins perform much better than serial joins. From the experimental results, we can see some interesting phenomena. In data mining applications, multi-label classification is in great demand for many modern applications. A useful data mining approach, on the other hand, is the k-nearest-neighbor join. This is more accurate, but takes longer to process. With the recent explosion of big data, traditional ANN join-based multi-label classification algorithms have to spend a lot of time processing large amounts of data. To address this problem, we first design his parallel MapReduce-based ANN join algorithm for big data classification. In his research, using Hadoop he further implemented the algorithm on a cluster of nine virtual machines. Experimental results show that his map-reduce-based ANN joins perform much better than serial joins. Several interesting phenomena are observed from the experimental results .

Josephine Usha Lawrance et al.,[21] In this study, a parallel clustering-based anonymization algorithm (PCAA) was proposed. The results prove that the algorithm is scalable and achieves a better tradeoff between privacy and utility. The MapReduce framework

is used to parallelize the anonymization process for processing large amounts of data. The algorithm performs well in terms of classification accuracy, F-measure, and Kullback-Leibler divergence metrics. In addition, big data generated from disparate data sources is efficiently protected to meet ever-growing application demands. In this research proposal, a parallel clustering-based anonymization algorithm (PCAA) was introduced to ensure privacy and usability of big data. The Hadoop MapReduce framework is used to parallelize the anonymization process for processing large amounts of data. The proposed big data clustering algorithms can be used to protect sensitive information in big data from various attacks. B. Combination attack, homogeneity attack, similarity attack, probabilistic reasoning attack. Based on several datasets of different sizes, the run-time efficiency and scalability of the proposed algorithm were investigated. Parallel clustering-based anonymization algorithms (PCAA) perform well in terms of F-value, classification accuracy, and Kullback-Leibler divergence metrics. Experimental results show that the proposed parallel clustering-based anonymization algorithm outperforms existing (G,S) and ANN-(G,S) approaches in terms of running time . This can be further improved by parallelizing the full clustering algorithm. This gives better results in terms of scalability and speed than future work. The proposed algorithms are designed to efficiently protect big data generated from disparate data sources, meet the demands of ever-growing applications, and ensure individual privacy before data is published and shared. We guarantee suitability.

Neha Bharill et al.,[22] The researchers focused on designing a partition clustering algorithm and implementing it in Apache Spark. In this work, we propose a partition-based clustering algorithm called Scalable Random Sampling with Iterative Optimization Fuzzy c-Means Algorithm (SRSIO-FCM) implemented in Apache Spark to address the challenges related to big data clustering. increase. To demonstrate the effectiveness of SRSIO-FCM compared to a scalable version of the proposed Literal Fuzzy c-Means (LFCM) called SLFCM implemented in Apache Spark, we performed experiments on several large datasets. The experiment will be carried out. Comparison results are reported in terms of F value, ARI, objective function, run time, and scalability. The reported results demonstrate the great potential of SRSIO-FCM for big data clustering. In this work, a novel SRSIO-FCM approach for scalable random sampling with iterative optimization and a fuzzy c-means approach for big data analysis were presented. SRSIOFCM processes big data piece by piece. A distinguishing feature of SRSIO-FCM is that it overcomes the problem of abrupt increases in the number of iterations that occur during clustering of subsets by feeding very different cluster centers generated from the previous subset as inputs. Subset. In this study, we applied SRSIO-FCM to four different datasets to demonstrate its feasibility and feasibility. In experiments performed on four datasets, SRSIO-FCM provides F-measures, ARI, objective function values, and significant improvements at runtime. Additionally, the study measures the scalability of SRSIO-FCM by varying the size of the data and the number of nodes in the cluster. Therefore, SRSIO-FCM is suitable for performing big data clustering in a time-efficient manner.

Engelbertus Vione et al.,[23] I am interested in observing the speed of performance using Hadoop to change the configuration of the number of slave nodes and applying Mahout's K-Mean algorithm to the UCI cluster liver disease dataset. For this study, he uses 4 computers with a configuration of 1 master node and 3 slave nodes in a Hadoop cluster running on a local network. In this study, a mahout library of 344 liver disease records was used to calculate several scenarios for the number of slave nodes in his Hadoop for K-Mean clustering. The research yielded a strategy for K-Mean computation using Hadoop consisting of PCs with i3 processors. Increasing the number of slave nodes from 1 to 3 increases computation speed non-linearly. For future directions, this work examines some computations of k-mean clustering using the mahout library using larger datasets.

Sami Al Ghamdi et al.,[24] Further improvements have been made to the efficiency and scalability of K-Means to deal with large data sets known as big data. In this work, we demonstrate K-Means optimization using the triangle inequality on two well-known distributed computing platforms, Hadoop and Spark. K-means variants that use triangular inequalities typically require caching additional information from previous iterations. This is a difficult task in Hadoop. Therefore, in this work, we present two methods of passing information from one iteration to the next in Hadoop to speed up K-Means. Experimental work shows that the efficiency of K-Means on Hadoop and Spark can be significantly improved by using triangle inequality optimization. Research improves the efficiency and scalability of K-Means. To achieve this goal, efficient variants of K-Means were implemented on Hadoop and Spark. A variant used triangle inequalities to reduce the number of distance calculations at each iteration. Some of these variants required additional information from previous iterations that Hadoop did not support. Therefore, two techniques have been proposed, Extended Vector (EV) and Bounds Files (BF), which allow him to pass the additional information that Hadoop needs from one iteration to the next. In addition, we examined the performance of several optimizations of K-Means on Hadoop and Spark. A comparative analysis of the EV and BF approaches showed that significant acceleration can be achieved by implementing both approaches. However,

implementations using BF are more efficient and scalable than implementations using EVs to pass information to subsequent iterations. The overhead of writing EVs to HDFS increases dramatically as the number of clusters and dimensions increases.

Daniel Peralta et al.,[25] An evolutionary computation-based feature selection algorithm that uses the MapReduce paradigm to retrieve feature subsets from large datasets. The algorithm decomposes the original data set into blocks of instances and learns from them during the map phase. The reduction phase then merges the obtained partial results into a final vector of feature weights. This gives you the flexibility to apply feature selection methods using thresholds to determine the subset of selected features. Feature selection methods are evaluated using three well-known classifiers (SVM, Logistic Regression, and Naive Bayes) implemented in the Spark framework to address big data problems. Experiments managed datasets with up to 67 million instances and up to 2,000 attributes. This demonstrates that it is a suitable framework for performing evolutionary feature selection and improving both classification accuracy and runtime when dealing with big data problems that require improvement. In this work, MR-EFS proposes an evolutionary feature selection algorithm based on the MapReduce paradigm to preprocess large datasets to make them affordable for other machine learning techniques such as bottom. B. Classification techniques that currently do not scale well enough to handle such data sets. The algorithm was implemented using Apache Hadoop and applied to two different large datasets. The resulting reduced dataset was tested with his three different classifiers implemented in Apache Spark on a cluster of 20 computers. A theoretical evaluation of the model highlights the full scalability of MR-EFS. Sequential procedure. This behavior was further confirmed after empirical procedures. According to the obtained classification results, it can be argued that MR-EFS can adequately reduce the number of features for large datasets. The result is a reduced version of them that is smaller to store and faster and easier to compute. Classify. These facts were observed for two different data sets and for all classifiers tested. For the Epsilon dataset, the reduced dataset size to node count ratio forces the HDFS block size to prove optimal utilization of Hadoop and Spark hardware resources with good design.

Omkaresh Kulkarni et al.,[26] I have a strong interest in traditional data processing techniques that extract hidden patterns and correlations from vast amounts of data known as big data. Clustering methods play an important role in reducing computational complexity. With knowledge of clustering algorithms, big data arriving from distributed sources are processed by the MapReduce framework (MRF). Since MRF has two functions, a mapping function and a reduction function, the mapping function is based on the proposed Fractional Sparse Fuzzy C-Means (FrSparse FCM) algorithm and the reduction function is based on the particle swarm-based Whale optimization algorithm. It is based on. Optimization (P-Wal). First, the best centroid is calculated using the proposed algorithm in the mapper stage and optimally adjusted in the reducer stage. It is clear that the proposed FrSparse-FCM-based his MRF guarantees parallel processing of big data. In this work, we focused on clustering big data using MRF based on the proposed FrSparse-FCM algorithm. First, the mapper in the mapper phase computes the optimal centroid using the proposed FrSparse FCM algorithm, and the reducer in the reducer phase performs classification using the P-Whale optimization algorithm. The developed FrSparse FCM integrates the fractional concept into sparse FCM, so the developed algorithm performs big data clustering with better classification accuracy. The optimal centroids determined using FrSparse FCM proposed in the mapper stage are optimally adjusted in the reduction stage to represent better centroids in the reduction stage. The importance of the proposed method is its ability to process big data from distributed sources. Experimentation is performed using the Skin dataset and localization dataset taken from the UCI machine learning repository, and the analysis is progressed using the metrics, such as accuracy and DB Index.

J.V.N. Lakshmi et al.,[27] has identified a difficult problem for network-centric applications that need to process large data sets. Systems need advanced tools to analyze these datasets. The programming models Map Reduce and Hadoop are used for extensive data analysis as efficient parallel computing. However, Map Reduce still has performance issues. Map Reduce uses individual shuffle service components in the shuffle phase with efficient I/O policies. The map phase needs performance improvements because the output of this phase serves as the input for the next phase. Its result reveals the efficiency, so map phase needs some intermediate check points which regularly monitor all the splits generated by intermediate phases. This is an obstacle to effective utilization of resources. In this study, shuffle is implemented as a service component to reduce overall job execution time, monitor the map phase through skewing, and increase resource utilization within the cluster. The map-reduce programming model requires improvements in both the map and shuffle phases. Although simple, implementation studies face complex problems at the card level. If the mapping fails, the output cannot be computed because the result of the mapping phase is the output of the reduce phase. During the reduce phase, a new scheduler is added with exactly one reducer assigned to each node. Using a Generate

function that dynamically monitors the Reduce phase solves the underlying problem of the Map phase. Shuffle as a Service can make use of cluster resources, and consumes less time over time when processing large amounts of intermediate data. Hadoop MapReduce jobs were used to demonstrate the performance benefits of Hadoop for different sizes, word count performance, and terasort. The shuffle service makes the best use of cluster resources and can process large amounts of intermediate data in a short time. Our skewing guidelines ensure high resource utilization and improve completion times for jobs that contain skewed tasks.

Simone A. et al.,[28] We studied the parallelism and scalability of a popular and effective fuzzy clustering algorithm called Fuzzy C-Means (FCM) algorithm. This algorithm is parallelized using the MapReduce paradigm, which outlines how the Map and Reduce primitives are implemented. A plausibility analysis is performed to show that the implementation works correctly and yields results of competitive purity compared to state-of-the-art clustering algorithms. Additionally, a scalability analysis is performed to demonstrate the performance of the parallel FCM implementation as the number of compute nodes used increases. Current clustering algorithms cannot handle big data, so a scalable solution is needed. Fuzzy clustering algorithms have shown to outperform hard clustering algorithms. Fuzzy clustering assigns membership degrees between 0 and 1 to the objects to indicate partial membership. This research investigated the parallelization of the FCM algorithm and outlined how the algorithm can be parallelized using the MapReduce paradigm that was introduced by Google. Parallelization requires two MapReduce jobs because the centroids must be computed before computing the membership matrix. The accuracy of the MR-FCM algorithm was measured in terms of purity and compared with different clustering algorithms (both hard clustering and fuzzy clustering techniques) with comparable results. This comparison shows that KMeans, the algorithm with the lowest computational complexity, performs the best. Although FKM and MR-FCM are computationally very similar, the mahout-FKM algorithm has been found to scale better than the MR-FCM algorithm. Overall, an implementation study showed how to parallelize his FCM algorithm using the MapReduce framework, and an experimental evaluation showed that comparable cleanness results could be achieved. Furthermore, the MR FCM algorithm scales well with increasing data set sizes as shown by the scalability analysis conducted. Future work includes applying the MR-FCM algorithm to different clustering data sets emphasizing on the purity and scalability. Furthermore, larger data set sizes containing GBs of data should be investigated. For this however, another Hadoop cluster needs to be utilized where big data sets can be processed to achieve larger data clustering.

Yaminee S et al.,[29] We presented a K-Means clustering algorithm in a distributed environment using Apache Hadoop. The main focus of this work is the implementation of the K-Means algorithm and the design of the mapper and reducer routines described in the work. The procedure for running the K-Means algorithm is also described in this research and serves as a guide for practical implementation. Data mining is one of the most important tools when gathering information. The amount of information exchange is increasing at a staggering rate, requiring the processing of enormous amounts of data. The research discusses the implementation of the K-Means clustering algorithm on distributed networks. This algorithm not only provides a robust and efficient system for grouping data with similar characteristics, but also reduces implementation costs for processing such large amounts of data. In this study, we investigated an important feature of data mining, namely cluster analysis using the K-Means algorithm. Theoretical studies and empirical examples show that K-means clustering with MapReduce is better suited for both text and web documents. This research focused on the K-Means clustering algorithm in a distributed environment using Apache Hadoop. In the future, the Hadoop framework can be used to implement various clustering algorithms to reduce operational resources and improve operational speed.

Gothai E et al.,[30] proposed a novel distributed supervised machine learning algorithm based on the MapReduce programming model and Distance Weighted k-Nearest Neighbor algorithm called MR-DWkNN to process and analyze the Big Data in the Hadoop cluster environment. The proposed distributed algorithm is based on supervised learning performs both regression tasks as well as classification tasks on large volume of Big Data applications. Three performance metrics, such as Root Mean Squared Error (RMSE), Determination coefficient (R2) for regression task, and Accuracy for classification tasks are utilized for the performance measure of the proposed MR-DWkNN algorithm. The extensive experimental results shows that there is an average increase of 3% to 4.5% prediction and classification performances as compared to standard distributed k-NN algorithm and a considerable decrease of Root Mean Squared Error (RMSE) with good parallelism characteristics of scalability and speedup thus, proves its effectiveness in Big Data predictive and classification applications. In this research have developed a MapReduce based on two different versions of the kNN algorithm called MR-SDkNN based on standard k-NN algorithm and MR-DWkNN based on distance weighted k-NN algorithm. The prediction and classification performance of MR-DWkNN is evaluated with three metrics: root mean square error (RMSE), coefficient of determination (R2), and accuracy. Moreover, the scalability performance of the proposed

algorithm was also tested on a Hadoop multi-node cluster. The results obtained from these experiments indicate that the main outcome of MR-DWkNN is the improved performance such as classification accuracy and coefficient of determination (R2) of the proposed MR-DWkNN compared to MR-SDkNN and MR- It was shown that there is DWkNN is a scalable approach in multi-node cluster environments and a proven parallel approach with promising performance metrics in big data applications. Future work under consideration is to use other big data processing frameworks such as Spark and Flink to improve runtime execution of Map and Reduce tasks.

A L Ramdani et al.,[31] We have discussed the analysis and implementation of the Pillar K-Means algorithm on distributed systems using the MapReduce framework. In this research, we took the existing Pillar K-Means algorithm and implemented it using the MapReduce framework. Various features are implemented at the same time, such as mappers and reducers that are part of the MapReduce framework. The results showed positive performance in terms of efficiency and scalability of the Pillar-K means by using synthetic datasets. Clustering algorithm is one of his unsupervised learning algorithms, which are widely used in various fields. In this work, the development of a pillar k-means algorithm running on the MapReduce framework was proposed. Hadoop was used in the algorithm development process. Results from two scenarios showed that using the MapReduce framework with the pillar k-means algorithm can significantly improve computational speed by increasing the number of nodes. Also, using Hadoop configuration to determine the appropriate number of card functions can improve computational speed when the number of core processors is the same as the number of card functions. However, this research needs to be investigated more closely, especially manipulating resources such as CPU and disk I/O to observe energy consumption.

Tanvir H et al.,[32] suggests that traditional clustering algorithms should be redesigned for modern computer architectures. This wok proposed a new MapReduce-based fuzzy C-means algorithm for clustering large document data. The algorithm has been extensively experimented with document records of different sizes and run on Hadoop clusters of different sizes. The efficiency of the proposed algorithm is compared with serial traditional fuzzy C-means and MapReduce-based K-means algorithms. The proposed fuzzy c-means algorithm design scaled well with the Hadoop platform, documented large datasets, and yielded improved performance. In the era of big data, new architectures require novel designs of traditional clustering algorithms to achieve efficient clustering times. A MapReduce-based modification of the clustering algorithm was recently developed and tested on the Hadoop platform. The research was developed and experimented with MapReduce-based fuzzy C-Means for big data documents. A wide range of experiments were performed on five different types of datasets and five different sizes of Hadoop clusters. Experimental evaluation shows that the proposed algorithm is scalable on his Hadoop and efficiently holds clustering jobs. Analysis of experimental data shows that Hadoop cluster overhead and algorithm design lead to uneven performance gains in terms of cluster size and dataset size. We have also found that large datasets and large Hadoop clusters are more effective in achieving performance gains. The main contribution of this paper is to test the proposed algorithm on document-type big data for clustering.

Xiaoli Cui et al.,[33] We address the problem of processing large datasets using the K-Means clustering algorithm and propose a new processing model in Map Reduce to eliminate iterative dependencies and achieve high performance. Research analyzes and implements our ideas. Extensive experiments on our cluster show that the proposed method is efficient, robust and scalable. In this study, iteration of the K-Means algorithm was shown to be a key factor affecting clustering performance, and a new efficient parallel clustering model was proposed. Experimental results on large real and synthetic datasets show that the optimized algorithm is efficient and outperforms parallel K-means, Kmeans and standalone Kmeans++ algorithms . Clustering validation shows that the quality of the clustering method is comparable to that of K-means.

Georgios et al.,[34] FML-kNN is a new distributed processing framework for big data that performs probabilistic classification and regression, implemented in Apache Flink. The core of the framework, unlike similar approaches, consists of a k-nearest neighbor join algorithm that runs in a single distributed session and can operate on very large datasets of varying granularity and dimensionality. In this study, we evaluate the performance and scalability of FML-kNN in a detailed experimental evaluation and compare it to similar methods implemented in Apache Hadoop, Spark, and Flink distributed processing engines. The results demonstrate the framework's overall superiority in all comparisons performed. In addition, this study applies his FML-kNN to real household water consumption data in two motivating use cases for water demand management. In particular, this study focuses on predicting water consumption using one hour of smart meter data and extracting consumer characteristics from shower water consumption data. In this study, we further discuss the results obtained and demonstrate the potential of a framework for extracting useful knowledge. In this work, a novel distributed processing

framework was presented that supports probabilistic classification and regression approaches. Its core algorithm is an extension of the distributed approximation kNN implementation optimized for efficient operation on a single distributed session. Implemented using the scalable Apache Flink data processing engine. The study conducted detailed experimental evaluations to evaluate the framework in terms of clock completion time and scalability, and compared it to similar approaches based on Apache Spark and Apache Hadoop. Our framework outperformed all competing implementations. This is because optimizations and the ability to run in a single distributed session allow the same workload to run significantly faster and scale better for larger datasets. In addition, the study conducted two of his real-world case studies related to water consumption. This demonstrates the framework's potential in knowledge extraction tasks from very large amounts of data. As real-world data collection continues to evolve in all areas, extracting knowledge from everyday activities becomes a major challenge. The direction of future work is as follows. The research conducts extensive case studies on more datasets from different sources to determine the framework's ability to perform ad-hoc data mining tasks. In addition, research investigates its applicability to data stream mining applications where the input is a continuous flow of data sets. Finally, this work enhances the power of knowledge discovery frameworks by adding distributed machine learning approaches to increase their potential in the growing field of big data analytics.

Guodong Li et al.,[35] Aimed at the problems of initial points selection, outliers influence and cluster instability of traditional K-means algorithm in big data clustering, an MS-K Means algorithm based on MapReduce framework was proposed. The algorithm selected multiple non-outlier points as the candidate center points, and mean shifted the candidate center points, used the maximum minimum principle to select K initial center points from the candidate center points, and then executed the K-means algorithm to find the final center points. In order to improve the running speed of the algorithm for clustering large data sets, the algorithm used MapReduce framework to implement parallel computing on Hadoop platform. The experimental results showed that the parallel MS-K means algorithm is feasible in big data clustering, and the algorithm performs well in terms of performance, speed and stability. In order to solve the problems that the traditional K-means algorithm is susceptible to outliers, time-consuming and poor stability in the context of largedata, this research proposes an MS Kmeans algorithm, which determines outliers in high-dimensional space by using the Mean Shift algorithm to move candidate center points to data-dense regions, selects initial center points using the maximum-minimum principle in MapReduce framework to achieve parallelization, verify the advantages of the algorithm proposed in this research. Experimental results show that the clustering speed, performance and stability of the MS-Kmeans algorithm are effectively improved on large data sets.

Omkaresh Kulkarni et al.,[36] presented a technique called FPWhale-MRF for big data clustering using the MapReduce framework (MRF) by proposing two clustering algorithms. In FPWhale-MRF, the mapper function estimates cluster centroids using a fractional tangent spherical kernel clustering algorithm developed by integrating fractional theory with a tangent spherical kernel clustering approach. The reducer combines the mapper outputs to find the optimal centroid using her proposed Particle-Whale (P-Whale) algorithm for clustering. The P-Whale algorithm is proposed by combining the whale optimization algorithm and particle swarm optimization to achieve effective clustering and improve its performance. Two datasets, a localization dataset and a skin segmentation dataset. In this research paper, we present a method for big data clustering using MRF, called FPWhale-MRF, which is based on two clustering algorithms, FTSK and P-Whale. FTSK, employed in Mapper, is developed by integrating fractional arithmetic into his TSK clustering algorithm to find cluster centroids. Reducer, on the other hand, includes P Whale, an optimization-based clustering algorithm developed by modifying his PSO with WOA for optimal clustering. Therefore, the proposed FPWhale MRF method effectively performs big data clustering using the proposed clustering algorithm. Experiments are performed on two datasets, localization and skin segmentation, and the results are compared with those of existing techniques such as his MKS-MRF, K-means-MRF, FCM-MRF and KFCM-MRF. The performance of the proposed method is evaluated using two metrics: clustering accuracy and DB index. FPWhale-MRF was able to reach maximum accuracy. Therefore, we can conclude that the proposed FPWhale MRF method can effectively perform big data clustering with maximum clustering accuracy compared with existing comparative methods.

Mo Haia et al.,[37] comprised of three typical platforms for processing big data: Various parallel clustering algorithms are used to compare Hadoop, Spark, and DataMPI: equal K-implies, equal fluffy K-means and equal Overhang. Tests are performed on various text as well as numeric dataset and groups of various scale. Hadoop, Spark, and DataMPI were used in this study to analyze a variety of textual and numerical datasets using parallel K-means, parallel fuzzy K-means, and parallel canopy. According to the findings of the experiments, the following: 1) for the equivalent dataset, when the

quantity of hubs increments from 2 to 6, the speedup of every calculation is bigger than 1; (2) The memoryup of each algorithm is greater than 1 when the dataset and the number of nodes remain constant; 3) For the same data set, when each node has 4GB of memory, DataMPI outperforms Hadoop by 60% and Spark by 32%; 4) When clustering data, a cluster with six nodes and six gigabytes of memory per node should be selected in order to achieve high clustering performance.

III. CONCLUSION

Bunching is the approach to finding an associated event in the space of the sizes. The task of identifying the collections in high-dimensional datasets has remained challenging and time-consuming. Nearly all of the data from clustering methods have recently been used to solve problems with higher dimensionality. In order to achieve effective outcomes, these methods are combined with the optimization processes among them. In big data management and classification, the Map-Reduce technique is more effective than traditional data structure methods. In order to manage large data sets, various methods and algorithms are discussed in this paper. It demonstrates how these machine learning algorithms will respond to all Big Data framework challenges.

IV. REFERENCES

[1]. N. Zanoon, A. Al-Haj, S. M. Khwaldeh," Cloud computing and big data is there a relation between the two: a study", International Journal of Applied Engineering Research, Vol.12(17), (2017), pp.6970-6982.

[2]. T. C. Havens, J. C. Bezdek, and M. Palaniswami, "Scalable single linkage hierarchical clustering for big data," in Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on. IEEE, 2013, pp. 396–401.

[3]. H. Kalia, S. Dehuri, and A. Ghosh, "A Survey on Fuzzy Association Rule Mining," Int. J. Data Warehous. Min., vol. 9, no. 1, pp. 1–27, 2013. 7. O. Daly and D. Taniar, "Exception Rules Mining Based on Negative Association Rules," in Proceedings of the International Conference on Computational Science and Its Applications (ICCSA 2004), 2004, pp. 543–552.

[4]. M. Z. Ashrafi, D. Taniar, and K. A. Smith, "Redundant association rules reduction techniques," Int. J. Bus. Intell. Data Min., vol. 2, no. 1, pp. 29–63, 2007.

[5]. D. Taniar, W. Rahayu, V. C. S. Lee, and O. Daly, "Exception rules in association rule mining," Appl. Math. Comput., vol. 205, no. 2, pp. 735–750, 2008.

[6]. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)

[7]. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Mag. **17**(3), 37 (1996)

[8]. Ingersoll, G.: Introducing apache mahout. IBM developer Works Technical Library (2009)

[9]. Mikut, R., Reischl, M.: Data mining tools. Wiley Interdisc. Rev. Data Mining Knowl. Discov. **1**(5), 431–443 (2011)

[10]. Chen, H., Chiang, R.H., Storey, V.C.: Business intelligence and analytics: From big data to big impact. MIS Q. **36**(4), 1165–1188 (2012)

[11]. Dietrich, D., Heller, B., Yang, B.: Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Wiley, Hoboken (2015) Machine Learning and Big Data Processing: Review 477

[12]. Chopra, A., Madan, S.: Big data: a trouble or a real solution? Int. J. Comput. Sci. Issues **12**(2), 221 (2015)

[13]. Twardowski, B., Ryzko, D.: Multi-agent architecture for real-time big data processing. In: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 3, pp. 333–337. IEEE (2014)

[14]. Amatriain, X.: Mining large streams of user data for personalized recommendations. ACM SIGKDD Explor. Newsl. **14**(2), 37–48 (2013)

[15]. Richter, A.N., Khoshgoftaar, T.M., Landset, S., Hasanin, T.: A multi-dimensional comparison of toolkits for machine learning with big data. In: 2015 IEEE International Conference on Information Reuse and Integration (IRI), pp. 1–8. IEEE (2015)

[16]. A.Malcom Marshall, Dr.S.Gunasekaran," A Survey on Job and Task Scheduling in Big Data" Dept. of Computer Science & Engineering Coimbatore Institute of Engineering and Technology, Coimbatore, India.

[17]. Yongyi Li , Zhongqiang Yang, Kaixu Han,"K-Means Parallel Algorithm of Big Data Clustering Based on Mapreduce PCAM Method",journal of Engineering Intelligent Systems,2021, vol 29 no 6 November 2021

[18]. Lei Chen, Wei Lu, Liqiang Wang, Ergude Bao,"Optimizing MapReduce Partitioner Using Naïve Bayes Classifier", 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, 978-1-5386-1956-8/17 $31.00 © 2017 IEEE DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.138

[19]. F.Ouatik, M.Erritali, F.Ouatik, M.Jourhmane," Comparative study of MapReduce classification algorithms for students orientation ",the international workshop on websearch and data mining,2020,journal of elseveir

[20]. Xuesong Yan, Zhe Wang, Dezhe Zeng, Chengyu Hu, Hao Yao,"Design and Analysis of Parallel MapReduce based KNN-join Algorithm for Big Data Classification", TELKOMNIKA Indonesian Journal of Electrical Engineering Vol. 12, No. 11, November 2014, pp. 7927 ~ 7934, ISSN: 2302-4046

[21]. Josephine Usha Lawrance & Jesu Vedha Nayahi Jesudhasan, "Privacy Preserving Parallel Clustering Based Anonymization for Big Data Using MapReduce Framework", applied artificial intelligence 2021, vol. 35, no. 15, 1587–1620.Https://doi.org/10.1080/08839514.2021.1987709

[22]. Neha Bharill , Aruna Tiwari ,"Fuzzy Based Clustering Algorithms to Handle Big Data with Implementation on Apache Spark", 2016,IEEE computer society, 978-1-5090-2251-9/16 $31.00 © 2016 IEEE DOI 10.1109/BigDataService.2016.34

[23]. Engelbertus Vione , J.B. Budi Darmawan,"Performance of K-means in Hadoop Using MapReduce Programming Model", ICSTI 2018, October 19-20, Yogyakarta, Indonesia Copyright © 2019 EAI DOI 10.4108/eai.19-10-2018.2282545

[24]. Sami Al Ghamdi , Giuseppe Di Fatta ,"Efficient Clustering Techniques on Hadoop and Spark", Int. J. Big Data Intelligence, Vol. 0, No. x, 2018 ,

[25]. Daniel Peralta, Sara del Río,Sergio Ramírez-Gallego, Isaac Triguero, JoseM. Benitez, "Evolutionary Feature Selection for Big Data Classification: A MapReduce Approach", Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015,

[26]. Omkaresh Kulkarni ,"MapReduce framework based big data clustering using fractional integrated sparse fuzzy C means algorithm" ,journal of IET Image Processing, ISSN 1751-9659, 2019

[27]. J.V.N. Lakshmi ,"Data analysis on big data: improving the map and shuffle phases in Hadoop Map Reduce", Int. J. Data Analysis Techniques and Strategies, Vol. 10, No. 3, 2018

[28]. Simone A. Ludwig ,"MapReduce-based Fuzzy C-Means Clustering Algorithm: Implementation and Scalability"

[29]. Yaminee S. Patil, M. B. Vaidya ,"K-means Clustering with MapReduce Technique", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 11, November 2015, ISSN (Online) 2278-1021

[30]. Gothai E , Usha Moorthy"Map-Reduce based Distance Weighted k-Nearest Neighbor Machine Learning Algorithm for Big Data Applications",journal of research square, https://doi.org/10.21203/rs.3.rs-684319/

[31]. A L Ramdani, H B Firmansyah,"Pillar K-Means Clustering Algorithm Using MapReduce Framework"

[32]. Tanvir H. Sardar,Zahid Ansari ,"MapReduce-based Fuzzy C-means Algorithm for Distributed Document Clustering", J. Inst. Eng. India Ser. B, https://doi.org/10.1007/s40031-021-00651-0

[33]. Xiaoli Cui , Pingfei Zhu · Xin Yang ·,Keqiu Li , Changqing Ji ,"Optimized big data K-means clustering using MapReduce", J Supercomput DOI 10.1007/s11227-014-1225-7

[34]. Georgios Chatzigeorgakidis , Sophia Karagiorgou, Spiros Athanasiou , Spiros Skiadopoulos,"FML-kNN: scalable machine learning on Big Data using *k*-nearest neighbor joins" ,journal of big data, 2018

[35]. Guodong Li , Chunhong Wang,"Parallel MS-Kmeans clustering algorithm based on MapReduce", Journal of Springer nature,2020

[36]. Omkaresh Kulkarni, Sudarson Jena and C. H. Sanjay , "Fractional Fuzzy Clustering and Particle Whale Optimization-Based MapReduce Framework for Big Data Clustering",

[37]. Mo Haia,b, Yuejing Zhanga, Haifeng Lia ," A Performance Comparison of Big Data Processing Platform "Based on Parallel Clustering Algorithms,journal of Procedia Computer Science 139 (2018) 127–135