Comparative Analysis of Social Media Analytics in Bigdata Using Fuzzy C Mean (Fcm), K-Nearest Neighbour (Knn) And K-Means Algorithms

[1]Jayabharathi, [2]Dr. M. Logambal,
Research Scholar, Department of Computer Science, Vellalar College for Women, Thindal, Erode, Tamil Nadu,India.

Assistant Professor, Department of Computer Science, Vellalar College for Women, Thindal, Erode, Tamil Nadu,India.

Abstract

One of the sites that generates a lot of data is social media. Data retrieved from social media is used to examine how the community behaves and perceives a specific occurrence or phenomenon. The amount of data used in the real-time situation grows linearly over time. It was found that one of the biggest data-producing sources, including Blogspot, Twitter, Facebook, Wikimapia, AIM, and YouTube, was growing at an uncontrollable rate. MapReduce is used in the Hadoop framework to carry out the job of filtering and aggregation as well as to maintain the effective storage structure. In order to gain insights from these platforms' sentimental research, social media analytics tools and technologies are used. In this essay, we talk about big data and MapReduce, which is crucial for using big data to analyze social networking site issues. By combining Hadoop and MapReduce, machine learning algorithms can improve the efficiency of big data. K-Nearest Neighbor (KNN), Fuzzy C Mean (FCM), and K-means algorithms.

Keywords: Sentimental Analysis, big data, machine learning, social media, KNN, FCM, K-Means

I.INTRODUCTION

The sole purpose of social media analytics (SMA) is to gather useful data from various social media platforms and guide it toward various conclusions. The use of social media in the tech industry includes the use of various tools, technologies, and communication techniques. Blogspot, Twitter, Facebook, Wikimapia, AIM, YouTube, and many more are examples of social media websites. Research on social media has largely focused on three main areas up to this point: group/network analysis (which describes how a group of people interacts with one another and identifies dominant users), content analysis (which examines moods/sentiments and hot popular topics in social media), and prediction of real-world traits or events. [1]. Since public data is the foundation of today's social media networks, SMA's primary function is to analyze and gather it. It provides us with in-depth knowledge of user interests, aiding in market regulation. SMA is a simple and effective method for determining what the audience anticipates and desires from other contents, which are readily apparent in trending searches. SMA is an effective technique, but it is also much more difficult and important. The world of social media has seen tremendous development in recent years, and its industry as a whole.

This vast amount of data falls into the big data category. Social media can download and share data that can be used to train classifiers for sentiment analysis. Processing such a large amount of data for sentiment analysis is very time consuming for a single node. One way to obtain such large amounts of data is to use more powerful machines, such as supercomputers, to increase processing power, but these systems are

not only costly but also easily scaled. You can also I can't do it either. Another way to process such large amounts of data is parallel processing. However, parallel programming becomes tedious because parallelization introduces issues such as synchronization, resource allocation, resource sharing, and error handling. MapReduce is a programming model that hides the above issues from the user and provides an easy-to-use platform for processing large datasets on large scalable clusters [2]. MapReduce was originally developed by Google
exclusive technology. Hadoop is the most well-known open source version of MapReduce out of the many available. which consistently gather and keep a range of data produced by social Researchers in computer science, particularly in the area of human computer interaction, have been interested in studying emotions. They have also been studied in the behavior sciences and psychology. From an application perspective, the importance of emotions in written data is rising. Take affective computing, mining opinions, natural language interfaces, and market research as examples. Classifying and predicting whether a document reflects positive or negative sentiment using machine learning algorithms is frequently helpful. Unsupervised and supervised machine learning methods are two subcategories of machine learning. Each document in the training set is labeled with the proper sentiment by the supervised algorithm using a labeled dataset. Unsupervised learning, however, uses datasets that have not been labeled with the proper emotions [3]. Several algorithms, including KNN, FCM, and K-Means, have been used in the sentiment analysis area as part of the machine learning approach. There are many comparisons among these algorithms already accessible which works with features and characteristics. In this essay, we'll concentrate on these algorithms, evaluate them, and list the factors that determine which algorithm outperforms the others.

1.1.Sentimental analysis

An author's statement or view about a particular thing or aspect is referred to as their sentiment. Sentiment analysis is the process of analyzing, probing, and extracting users' preferences, sentiment, and opinions from subjective writing. The analysis of the text is the primary objective of sentiment analysis. Finding the text's valence is the simplest definition of sentiment analysis. Positive, negative, or neutral polarity can all be present. Because it gets the user's opinion, it is also referred to as opinion mining. Users have a variety of opinions, and sentiment analysis greatly aids in knowing them. Moods vary,

- **Direct opinion** The opinion about an item is expressed directly, as the name would imply, and it may be positive or negative. For example, "The cellphone's video clarity is poor" conveys a direct view
- **Comparison opinion,** Given that two identical things are being compared, it is a comparative statement. One potential example of conveying a comparative opinion is the statement, "The picture quality of camera-x is better than that of camera-y."

At various stages, sentiment analysis is carried out:

- Sentiment analysis determines whether a phrase is objective or subjective at the sentence level. Analysis at the sentence level makes the assumption that there is only one viewpoint present.
- Document-level sentiment analysis categorizes opinions about the specific object. The entire document contains opinions about the individual object and from the individual opinion maker.

1.2. Types of Sentiment analysis

- **Fine grained sentiment analysis:** Polarity accuracy can be crucial in certain circumstances when performing fine-grained mood analysis. Very positive, positive, neutral, negative, and very negative are additional polarity classifications that can be used. Commonly known as fine-grained mood analysis, this is. This can be used to understand a test's grade of five. I.e., very optimistic = 5; very negative = 1.Detection of feelings
- **Emotion detection:** This kind of mood analysis is used to identify feelings like joy, rage, sadness, and frustration, among others. The lexicon, or database of words

and the emotions they communicate, is used by the emotion recognition system. However, using lexicons has some restrictions. People have various methods of expressing their emotions. Words that are typically used to convey rage, such as "bad" or "kill," can also be used to convey joy.

- **Aspect based sentiment analysis :** It can be useful to know which specific aspects or features people are praising, criticizing, or ignoring when analyzing the emotions of text, such as policy reviews. Aspect-based mood analysis can be useful in these situations.
- **Multilingual sentiment analysis:** Multilingual sentiment analysis is a complex process. It requires a lot of pre-processing and resources. Most of these lexicons are available online (example mood lexicon). Others like translated corpora or noise detection algorithms need to be created.

II. MAP REDUCE

big files on HDFS can be handled using the MapReduce programming model, which allows the processing of big datasets using a parallel and distributed algorithm [4]. It is composed of the Map and Reduce operations and is carried out in three steps: Map, Shuffle, and Reduce. A worker processes the data and produces key-value pairs after using the map function to divide the incoming data into several manageable chunks. The Shuffle stage groups these combinations according to keys. Each group is delivered to the relevant Reducer. The Reducer then generates a fresh collection of output to be stored in HDFS. Distributed processes for both Map and Reduce are executed simultaneously. Huge data clusters can be processed by HDFS for use in high bandwidth user apps [5]. The user applications tasks are carried out by thousands of servers that are directly linked to the storage area and whose number is easily expanded in response to user demands. The MapReduce paradigm offered by the Hadoop framework makes it simple to transform and analyze large data collections. Hadoop's ability to distribute data and calculations across tens of thousands of computers while simultaneously running multiple apps is its most crucial feature.

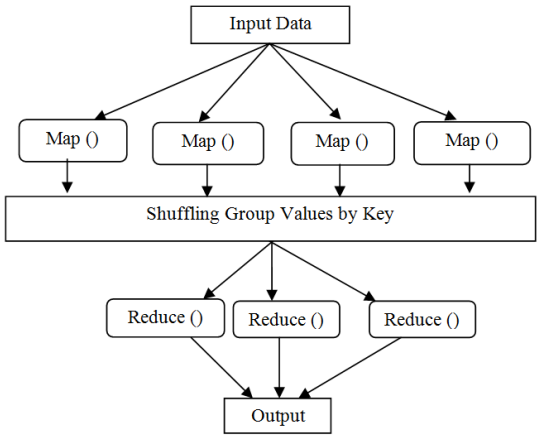

Figure 1: Map Reduce framework

A novel technology is needed for Big Data because of the difficulties that volume, velocity, and variety present. As the first practical platform for big data analytics, Hadoop is presently taking the lead in the field. The "three Vs" of Big Data can be managed by Hadoop, an open-source software architecture for scalable, dependable, distributed computing systems. MapReduce [6] and Google File System (GFS) [7] were the original sources of inspiration, Hadoop uses a straightforward programming paradigm to process large datasets across computer clusters and distribute memory. Because data processing is performed on a cluster of computers, it is likely that a node failure will occur during processing. Rather than relying on expensive, fault-tolerant servers, Hadoop manages node failures through its service, which can identify node failures in the cluster and distribute data to other available machines.

2.1 HADOOP

A distributed computing environment can handle huge amounts of data with the help of the Hadoop programming framework. We concentrate on Google's MapReduce framework's most widely used open-source version, Hadoop MapReduce [8]. To receive and write data, the Hadoop MapReduce framework makes use of a distributed file system. The open-source alternative to Google File System, Hadoop MapReduce typically makes use of Hadoop Distributed File System (HDFS) [9]. Large data sets can be processed using the

Hadoop programming model using a distributed computing system. The MapReduce component of the Hadoop ecosystem, along with the HDFS layer (Hadoop distributed file system), is what makes up the ecosystem. According to HDFS [10], this distributed file system is made to operate on top of the local file system of the cluster node and store extremely large files that are suitable for streaming data access. HDFS can scale up from a single server to hundreds of machines and is extremely fault tolerant.

A. Modules of Hadoop Framework

Hadoop common: This directory includes libraries and utilities that are required by other Hadoop modules. HDFS, or Hadoop Distributed File Systems: a distributed file system with very high aggregate bandwidth across clusters that saves data on common machines.

1. ***Hadoop Yarn:*** It is platform used for managing clusters of computing resources and scheduling user apps.

2. ***Hadoop Map Reduce:a*** map-reduce programming model implementation for handling huge amounts of data.

The computation is divided into two stages by the MapReduce computing model: Map and Reduce. The master node divides the incoming dataset into separate sub-problems during the Map phase and assigns each sub-problem to a worker node. Then, the worker nodes handle the smaller problems in parallel and send their findings back to the master node. The master node takes the solutions to each sub-problem and combines them in some way during the reduce phase to create the output. Users in this paradigm only need to define what to compute in the map and reduce functions, while the system automatically distributes data processing across a highly distributed cluster of machines. In the MapReduce model, all computation is organized around pairs. In the first stage, the map function takes a single pair (key, value) as input and produces a list of intermediate pairs (key, value) as output. It could be represented as:

$$\text{Map}(\text{key1},\text{value1}) \rightarrow \text{list}(\text{key2},\text{value2})$$

Then the system merges these intermediate pairs and groups them by key and passes them to the reduce function. Finally, the reduce function takes a key and an associated list of values as input and generates a new list of values as output, which can be represented as follows:

$$\text{Reduce}(\text{key2},\text{list}(\text{value2})) \rightarrow (\text{key2},\text{value3})$$

III.MACHINE LEARNING APPROACHES

Machine learning is a technique that builds models to autonomously analyze the data. Text classification involves training models on a particular set of documents to take advantage of patterns in natural language text vector representations. The attribute values of the class identifier for the record's class are known in advance for collections of papers or records. Training data is what it is termed. After training is finished, one applies the model to a new data collection known as test data to assess how well it performed. Sometimes portion of the whole dataset is also used as a validation dataset for model selection. Clustering, classification, regression, and other categories are used to group machine learning methods. The following provides an explanation of the various text classifications found in social media data. how many steps are typically needed in a machine learning technique. It is possible to describe how this method functions as follows:[11][12]

- • **First Step:** Data Collecting - Depending on the application, data is gathered at this point from a variety of sources such as blogs and social networks (Twitter, MySpace, etc.).
- **Second Step: Pre-processing –** The acquired data is cleaned and prepared for feeding into the classifier at this point. Removal of symbols and phrases is part of cleaning. Emoticons, for instance, are smileys used in text to express feelings. Changing all capital and lowercase to the standard case, eliminating non-English (or suggested language) texts, eliminating unnecessary white spaces and tabs, and so on.
- **Third Step: Training Data –** The most popular form of crowd-sourcing creates a hand-tagged compilation of data. This information will be given to the algorithm for learning purposes as the classifier's fuel.
- **Fourth Step: Classification –** The entire method is built around this. SVM or Naive

bayes are used for research, depending on the needs of the application. The classifier is prepared to be applied to real-time tweets and text for sentiment extraction after training.

- **Fifth Step: Results –** According to the chosen representation style, such as charts, graphs, etc., the results are plotted. Before the code is released, performance tuning is performed.

IV. CLUSTERING ALGORITHMS

A clustering algorithm is used to identify any groupings in a large amount of input data that do not have labels. Clusters are what make up those collections. A cluster is a collection of data points that are related to one another based on how they relate to other data points in the area. The characteristic development techniques listed below, to feed example, use clustering.

4.1. K-Means

Hard, K-Means As a partitioning technique used to analyze data, C-Means clustering considers observations of the data as objects based on their locations and the distances between different input data points. It divides the objects into mutually exclusive clusters (K) while keeping objects in each cluster as near as possible to one another and as far away as possible from objects in other clusters. Each cluster is characterized by its centre point i.e. centroid. The distances used in clustering in most of the times do not actually represent the spatial distances. In general, the only solution to the problem of finding global minimum is exhaustive choice of starting points. But use of several replicates with random starting point leads to a solution i.e. a global solution [13][14][15], In a dataset, a desired number of clusters K and a set of k initial starting points, the K-Means clustering algorithm finds the desired number of distinct clusters and their centroids. The point whose coordinates are determined by averaging each of the coordinates of the points of samples given to the clusters is known as a centroid. Due to the iterative algorithm used to determine the minimum sum of distances between each object and its cluster centroid across all clusters.A given data set can be categorized using a priori fixed defined number of clusters using the k means algorithm.The main goal is to locate k clusters, and centroids must be positioned deftly because various distance clusters yield different outcomes.These centroids ought to be picked with care, since various locations will have distinct consequences. The best option is to situate them as far apart from one another as you can. The next stage is to take each point from a given data set and associate it with the closest centroid. The first step is finished and an early grouping is made when no further points are discovered to be pending. We now need to compute k new centroids of the clusters that were produced in the previous step once more. Now that we have k new centroids as a result of all this, a new binding must be created between the same data set elements and the closest new centroid. There has been created a cycle. Because of this cycle, we might be able to detect the end of clustering when the k centroids stop gradually shifting their positions. To put it another way, centroids are no longer moving, and the primary goal of K means is to find clusters of data within data that meet the stopping requirements.

Algorithm: Standard K-means
Input: Dataset D={x1,….,xn} and K (number of clusters to be generated)

Step 1: Select centers C randomly from D
Step 2: If u > iter then /* loop repeats until convergence
Step 3: for each xi compute distance from all centers Cj
Dist (xi, Cj)
Step 4: Assign xi to closest Cj (min(dist(xi, Cj))
Step 5: Calculate new centroid

$$n_i = \frac{1}{c_i}\sum_{i=1}^{c_i} x_i$$

Step 6: end for
Step 7: End if
Step 8: Exit

This is a standard method of clustering datasets using K-means algorithm. However, there are a few issues with the method [16][17]:

1. The centroids' initialization procedure is not defined. It is common to select K examples at random.
2. The original centroid values determine the clustering quality, and it is possible to repeatedly find suboptimal partitions. The only solution that is frequently used is trying a variety of various starting points.

3. K's value impacts the outcome.
4. The common algorithm is straightforward, but it takes a long time to complete when dealing with big, multidimensional datasets. In this case, a single machine's memory might not be adequate.

One of the supervised learning methods, the K-Nearest Neighbor algorithm is frequently used for estimation and prediction in addition to classification and pattern recognition. The K-Nearest Neighbor algorithm uses memory, doesn't require a special model to fit it, and has a straightforward idea. Other than gathering vectors labeled with the specified class, no specific training procedure is required for a set of observations. Finding the closest k value in the training set, looking for the most votes in iteration k, and class labeling in each classification are all intensive computations that are done on classifications that involve two observations. [18]. $(pp, qq) =)*(qq+ - pp+)- . +/0$ (1) where i is iteration, n is the utmost iteration, p is input data, q is training data, and d is distance. The procedures used in the K-Nearest Neighbor categorization are: 1. Establish the variable k. 2. Use the Euclidian technique to determine the distance between each data point in order to determine the distance as in equation 1. 3. Distance from small to large-based data sorting 4. Take the given data k. 5. Finding the data that has the greatest amount of determined k 6. Determine the class of data from the most number received. It is a member of the family of techniques known as lazy learning, which do not require an explicit training phase. This technique needs all of the data instances to be stored, and it classifies unknown cases by locating the class labels of the k instances that are nearest to them. Numerous distances or similarity measures can be computed to determine how similar two examples are to one another. To the full training dataset, this procedure must be applied to all input examples. Applying it in the setting of big data may compromise the response time as a result.

The nearest neighbor algorithm (k-NN)

There are two stages to the k-Nearest Neighbor Algorithm.

• The Preparation Phase
• The Evaluation Phase

The subsequent subsections go into great depth about these Phases.

1) This kNN's Training Phase No explicit training step is needed by the algorithm to classify the data. The class label and data vector coordinates are typically stored during the training process. The class name in general is used as an identifier for the data vector. During the testing process, this is utilized to categorize data vectors

2) The Evaluation Stage

Our goal is to identify the class name for the new point given the test data points. Following a discussion of the method for k=1 or the nearest neighbor rule, k=k or the k-Nearest Neighbor rule is introduced

a) k=1 or 1 Nearest Neighbor

- The easiest case for classification is this one. Let 'x' represent the labeled spot.
- In the training data collection, locate the point that is closest to 'x'. Let 'y' be the nearest point.Assigning the label of "y" to "x" is what the closest neighbor rule now requests. This strikes me as being overly simple and occasionally even illogical. Only when there are a small number of data points does this logic make sense. If there are many data sets, there is a very high probability that the labels for variables x and y will match. Consider the scenario where we are handed a (possibly) biased coin. If you throw it a million times, you get a cranium 900,000 times out of a million. The subsequent decision will then most likely be a head. Here, a similar reasoning is valid.

b) k=K or k-Nearest Neighbor

This is a natural expansion of 1NN. A given data point from the testing data set is identified by its 'k' nearest neighbors, and its classification is determined by a majority decision in the training data set. When there are two divisions, "k" is typically odd. The majority voting is demonstrated in the following case. Think about a data point from the testing data collection. In the training data collection, there are 3 data points with the class label C1 and 2 data points with the class label C2 close to the data point that needs to be classified. Let's say that k = 5. In this instance, kNN predicts that the new data point, which

constitutes the plurality, will be labeled as C1. When there are numerous classes, a similar argument.

3) Generalising the k-Nearest Neighbor Testing Phase

Find the number of "k" in 1. (input)
2. Prepare the training data collection by logging the data points' coordinates and class labels.
3. Load the test data point.
4. Use a distance metric to conduct a majority vote among the 'k' nearest neighbors of the testing data point in the training data collection.
5. Label the newly added data point from the test data collection with the class label of the majority vote winner.
6. Carry on in this manner until all the testing phase's data elements are classified.

Algorithm 1: The k-nearest neighbors (KNN) algorithm.
Data: D = f (xi, ci), f or i = 1 to mg, where xi = (vi1, vi2, ..., vin) is an observation that belongs to class ci
Data: x = (v1, v2, ..., vn) data to be classified
Result: class to which x belongs
Distances Æ;
For yi in D **do**
Di ← d(yi, x);
Distances ← distances U {di};
End

Sort distances = fdi , f or i = 1 to mg in ascending order;
Get the first K cases closer to x, DKx ;
class most frequent class in DK

KNN classifies an instance by finding its nearest neighbors [19]. The KNN classifier applies the Euclidean distance or cosine similarity for differentiating the training tuple and test tuples. The same Euclidean distance between tuples Xi and Xt (t = 1,2,3 …n) can be explained as[20]:

$$d(y_i, y_t) = \sqrt{(y_{i1} - y_{t1})^2} + (x_{i2} - xt2)^2 + ..., (x_{is} - x_{ts})^2 \quad (2)$$

where yi, n and s be the tuple constant. It can be written as:

$$Dist(y1, y2 = \sqrt{\sum_{i=1}^{n} (y_{1i} - y_{2i})^2} \quad (3)$$

This equation is prepared based on KNN algorithms, every neighboring point that is closest to the test tuple, which is encapsulated and on the nearby space to the test tuple[21].

4.2. Fuzzy C-Means Algorithm

The FCM clustering algorithm, which is essentially based on Ruspini Fuzzy clustering theory, was proposed in the 1980s along with the development of fuzzy theory. Using the distance between different input data points as a basis for research, this algorithm is used. According to the separation between the data points, clusters are made, and for each cluster, cluster centers are created.The optimization of the fundamental c-means objective function or a variation of the objective function forms the basis of the majority of analytical fuzzy clustering methods. A number of techniques, such as iterative minimization, can be used to solve the nonlinear minimization problem represented by the optimization of the c means functional. The Fuzzy C-Means (FCM) algorithm, which uses the straightforward Picard repetition through the first-order conditions for stationary points, is the most widely used technique. The convergence of the FCM algorithm has been demonstrated by Bezdek in [22]. By incrementally lowering the weighted within group sum of squared error goal function, an optimal c partition is created:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} (u_{ij})^m d^{\ 2} (y_i, c_j)$$

Where Y = [y1, y2, ..., yn] is the data established in a d-dimensional vector space, n is the number of data items, c is the number of clusters which is defined by the user where $2 \leq c \leq n$, uij is the degree of membership of yi in the jth cluster, m is a weighted exponent on each fuzzy membership, cj is the center of cluster j, d2(yi, cj) is a square distance measure between object yi and cluster cj. Algorithm 1 describes an iterative procedure that can be used to reach an optimal answer with c partitions. The fuzzy partition matrix is initialized first, and then the subsequent actions are carried out in an iterative cycle: Equation 5 is used to

update the cluster centers, and Equations 6 and 7 are used to update the membership matrix. Up until a specific threshold number is achieved, the loop is repeated[23].

Algorithm:

Let X={x1,x2,x3,…,xn} be the set of data points and v={v1,v2,v3,…,vc} be the set of centers

1:Select "c" cluster center randomly

2:Calculate the fuzzy membership "uij" using:

$$\mu_{ij} = (\sum_{k=1}^{c} (\frac{d_{ij}}{d_{ki}})^{\frac{2}{m-1}}))^{-1} where \;\; d_{ij} = [x_i - v_j] \qquad (5)$$

Where: 'n' is the number of data points 'uij' represents the membership of ith data to the jth cluster center and

$$\sum_{j=1}^{c} u_{ij} = 1 \qquad (6)$$

'm' is the fuzziness index $m \in 1, \infty$

'dij' represents the Euclidian distance with between ith data and the jth cluster center

3. Calculate the fuzzy centers 'vj' using:

$$v_i = \frac{\sum_{i=1}^{n} (u_{ji})^m x_i}{\sum_{i=1}^{n} (u_{ji})^m} \qquad (7)$$

4. repeat steps2 and 3 until the minimum 'j' value is achieved ||U (l+1)-U (k)||< β is the termination criterion between [0,1].'U'=(μ_{ij}) n*c is the fuzzy membership matrix.'J' is the objective function

$$j(u,v) = \sum_{i=1}^{n} \sum_{j=1}^{c} (\mu_{ij})^m \| x_i - v_j \|^2 \qquad (8)$$

Where ||xi-vj|| is the Euclidian distance between ith data and jth cluster center

One of the most potent and well-known grouping algorithms is the FCM. Applying FCM to complex data grouping issues is inevitable. Parallel or distributed computing becomes a widely-used and reliable mechanism to address the problems that arise in many conventional machine learning techniques. An efficient parallel computing approach is provided by the Map-Reduce framework. With membership weights that have a natural interpretation but aren't at all probabilistic, FCM clustering methods, which are based on fuzzy behavior, offer a method that is natural for creating a clustering. which is used to locate cluster nodes that reduce a dissimilarity function. In order for the given information to belong to multiple groups, FCM employs fuzzy partitioning. A minimum of squared error immobile point of J within the following equation is a requirement for methods of minimization in order to achieve a high-quality classification. Using an iterative optimization of the objective, fuzzy portioning is applied. Both k means and FCM use the same clustering technique, but the k means algorithm clusters differently, In order to locate visible lumps or the cancer or tumor's starting point, the weighted cluster's mean is used. Each location in FCM is taken into account to have a weighted value linked to the cluster. This method aided physicians or radiologists in their efforts to determine the prevalence of breast cancer. On early cluster centers, the performance is based. It is challenging to distinguish the initial partitions in FCM because of the existence of outliers and noise. FCM performs better than the k-mean and KNN algorithms.

V. EXPERIMENTAL RESULT

Utilizing the effectiveness of KNN, K-Means, and FCM in MATLAB, Windows 10 OS, Intel i3 processor, and 4GB RAM, the trial is carried out. Sentiment140, 476 million Twitter Tweets, and Social Computing Data Repository are among the databases that are frequently used by machine learning researchers, particularly for the study of empirical algorithms in this field.

Accuracy: The accurate accuracy of classified instances with should be provided is calculated by dividing the number of classified items by the total number of classified items. Because it is a practical method to account for unbalanced test data in multiclass classification jobs, 100% accuracy results were chosen as the main criterion for comparing the effectiveness of different strategies. The FCM algorithm outperforms KNN and K-means by a significant margin and is comparable to the provided classifier for the given dataset. Figure 3 unequivocally shows that the FCM algorithm confirms superior classification accuracy when compared to other methods. Considering being true that the FCM method delivers outstanding outcomes on datasets with small sample sizes and that model creation time decreases as data volumes increase.

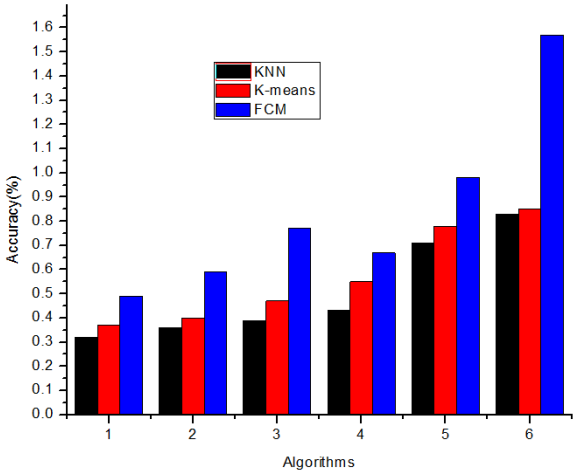

Figure 2: Accuracy

Execution Time: There are 6 test findings for the execution time analysis because time analysis was carried out three times and resulted in one graph for each analysis. Following that, the mean completion time for these three was determined. As a consequence, the FCM algorithm performed faster than the KNN and K-means algorithms. Figure 3 demonstrates that, despite a rise in data dimension, the execution time is significantly shorter. Nevertheless, the machine learning algorithm phase that analyzes customer behavior conducts numerous dot product operations, which makes it reliant on the dimension. Comparatively to the accuracy analysis findings, which are shown as results per figure3, the timing of the analysis was displayed as time-minutes graphs with the numbers of data.

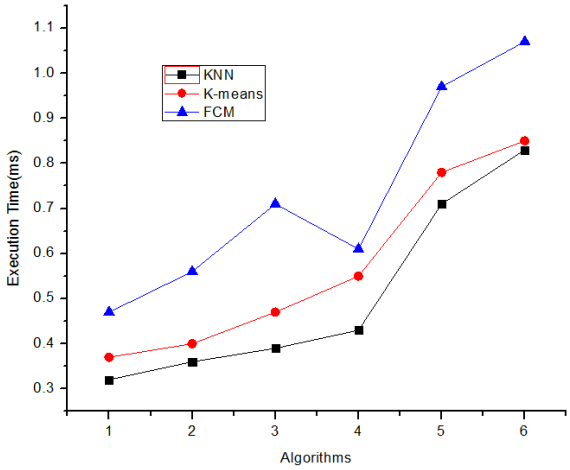

Figure 3: Execution Time

Memory Comparison: Figure 4 makes it obvious that the KNN algorithm uses the most memory at all level support because it generates candidate itemsets.

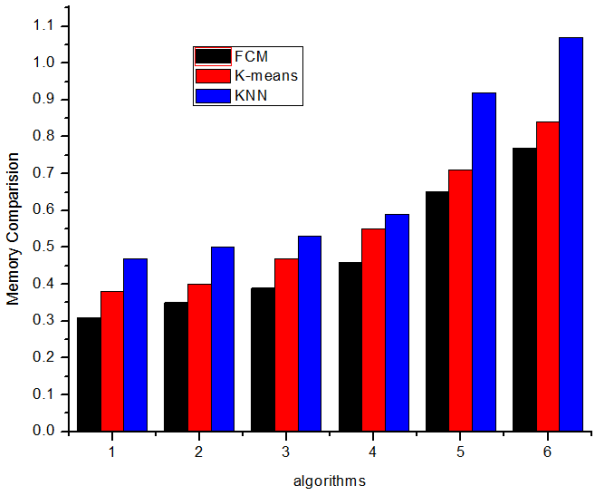

Figure 4: Comparison performance of FCM,KNN,K-means based on memory

At higher support levels, the memory usage for K-means and FCM approaches is roughly the same because as support increases, the likelihood of finding the maximal Itemset whose repetition is higher than the minimum support is decreased, making it more likely that both approaches will function similarly. When compared to KNN and FCM, the K-means approach works better. Since the FCM approach outperforms the K-means and KNN method, it can be concluded from the findings above.

Scaleup : When more iterations are available, the scaleup metric measures how well the parallel algorithm handles bigger datasets

$$scaleup = \frac{T_1}{\tilde{T}_p}$$

where T1 refers to the algorithm's sequential execution time for processing the provided dataset on a single node and T_p denotes the algorithm's parallel execution time for processing p-times larger datasets on p-times larger nodes. We increase the amount of data (from one iteration to six iterations) in direct proportion to the size of datasets to validate the scaleup of FCM, and then plot the outcomes in Figure 5. Figure 5 shows that all scaleup values of FCM are larger than 0.47, with the number of iterations and dataset size increasing proportionally.. The results indicate that the proposed FCM algorithm on MapReduce scales very well and has the adaptability of big datasets compare to other algorithms.

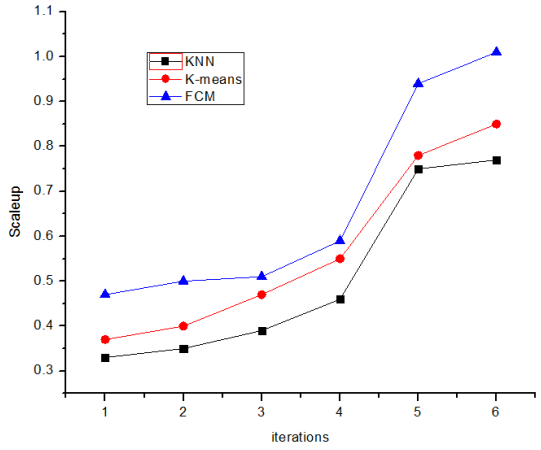

Figure 5: FCM compare with other algorithms using Scaleup

VI. CONCLUSION

Traditional data processing software cannot handle big amounts of data. Big Data is a novel term in this context. A huge dataset is referred to as big data. Big Data makes use of a novel tool known as Hadoop. It is free and open source software for scalable and distributed processing. Using basic programming models, this software enables for the distributed processing of datasets across clusters of computers. It can grow from a single server to thousands of machines. Each provides computation and storing on a local level. Hadoop makes use of the Map Reduce method. Using a machine learning algorithm called Map Reduce, this study analyzes sentiment in social media using big data. The FCM technique demonstrates that it is highly scalable and provides greater accuracy with a given dataset. According to the experimental findings, the FCM algorithm appears to be superior to the other algorithms for social media datasets. Twitter ratings are provided by the social networking site.

VII.REFERENCES

[1]. Gastelum, Z. N., & Whattam, K. M. (2013). State-of-the-art of Social Media Analytics Research. https://doi.org/10.2172/1077994

[2]. J. Dean and S.Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, in Communications of the ACM 51.1, Jan. 2008, pp. 107-113

[3]. Y. Singh, P. K. Bhatia, and O. Sangwan, “A review of studies on machine learning techniques,” International Journal of Computer Science and Security, vol. 1, no. 1, pp. 70–84, 2007.

[4]. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. Ieee, 2010, pp. 1–10.

[5]. J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[6]. Ha, I. , Back, B. , & Ahn, B. (2015). MapReduce functions to analyze sentiment information from social big data. International Journal of Distributed Sensor Networks , 11, 417502.

[7]. S. Ghemawat, H. Gobioff and S. Leung. 2003. “The Google file system”. *Proceedings of the nineteenth ACM symposium on Operating systems principles*.

[8]. J.Dean and S.Ghemawat, Mapreduce: a flexible data processing tool CACM, 53(1): 72-77, 2010.

[9]. K, shvachko, K. Haronz, S .Radia, R. Chansler, The Hadoop Distributed File System, Mass storage systems and Technologies(MSST), 2010 IEEE 26th Symporuenson, 2010, PP-1-10.

[10]. A. Asuncion and D. J. Newman, UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science, 2013.

[11]. Mathworks. http: //www.mathworks.com

[12]. A. K. Jain, M. N. Murty and P. J. Flynn, “Data Clustering: A review”, ACM Computing Surveys, vol. 31, no. 3, 1999.

[13]. J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, 2nd ed., New Delhi, 2006

[14]. X. Hui, J. Wu and C. Jian, “K-Means clustering versus validation measures: A data distribution perspective”, IEEE Transactions on Systems, Man, and cybernetics, vol. 39, Issue-2, 2009 , pp.319-331.

[15]. Pandove D and Goel S. "A comprehensive study on clustering approaches for big data mining," *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, Coimbatore, 2015, pp.1333-1338. doi: 10.1109/ECS.2015.7124801

[16]. Qiao J and Zhang Y. "Study on K-means method based on Data-Mining," *2015 Chinese Automation Congress (CAC)*, Wuhan, 2015, pp. 51-54. doi: 10.1109/CAC.2015.7382468

[17]. Ouyang, J.; Luo, H.; Wang, Z.; Tian, J.; Liu, C.; Sheng, K. FPGA implementation of GZIP compression and decompression for IDC services. In Proceedings of the 2010 International Conference on Field-Programmable Technology, FPT'10, Beijing, China, 8–10 December 2010; pp. 265–268.

[18]. K. Saxena, Z. Khan, and S. Singh, "Diagnosis of Diabetes Mellitus using K Nearest Neighbor Algorithm," vol. 2, no. 4, pp. 36–43, 2014.

[19]. G. G. Várkonyi and A. Gradišek, "Data protection impact assessment case study for a research project using artificial intelligence on patient data," *Inform.*, vol. 44, no. 4, pp. 497–505, 2020.

[20]. https://doi.org/10.31449/INF.V44I4.3253

[21]. S. K. Nayak, M. Panda, and G. Palai, "Realization of optical ADDER circuit using photonic structure and KNN algorithm," *Optik (Stuttg).*, vol. 212, no. March, p. 164675, 2020.

[22]. J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers Norwell, MA, USA, 1981.

[23]. Simone A. Ludwig, Clonal selection based fuzzy C-means algorithm for clustering, GECCO '14 Proceedings of the 2014 conference on Genetic and evolutionary computation, Pages 105-112.