Iranian Journal of Fuzzy Systems

Volume 21, Number 2, (2024), pp. 141-160

Original paper

University of Sistan and Baluchestan

Advancing big data clustering with fuzzy logic-based IMV-FCA and ensemble approach

L. S. Dandugala [1] and K. S. Vani [2]

[1] *Research Scholar, Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India*
[2] *Professor, Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Kanuru, Vijayawada, Andhra Pradesh*

dandugalalakshmisrinivasulu@gmail.com, vanikonerusuvarna@gmail.com

Abstract

The act of gathering, looking over, and analyzing a lot of data to find patterns, insights, and market trends that can help businesses make more effective choices is known as big data analysis (BDA). Quick and effective access to this data allows businesses to be flexible in developing strategies to hold onto their competitive edge. To analyze massive amounts of data quickly through parallel processing, the structure of the Hadoop software employs the MapReduce methodology. Computational solid resources are necessary for BDA, although they are not always available. Developing new clustering techniques that could handle this kind of data processing became crucial. Therefore, in this research, we presented a novel, effective fuzzy-based Improved Multiview Fuzzy C-Means Algorithm (IMV-FCA) to boost the clustering strategy. To summarize, fuzzy-based IMV-FCA clustering presents the ensemble of the MobileNet V2 model, and three-layered stacked Bidirectional LSTM (MVSBiLSTM) to increase computing speed and effectiveness. It also presents a function that calculates the separation among the cluster center and the particular instance, to assist with better clustering. By simulating shared memory space and parallelizing on the framework known as MapReduce on the Hadoop cloud computing platform, the distributed database is utilized to improve the method's effectiveness while reducing its time complexities. The experimental investigation was conducted on existing approaches, and the proposed approach was analyzed using three standard datasets. While differentiating from existing approaches, the presented approach yields greater performances in terms of various metrics.

Keywords: Big data analytics (BDA), Hadoop, cloud computing, improved multiview fuzzy C-means algorithm (IMV-FCA), MobileNet V2, three layer stacked Bidirectional LSTM, mapreduce.

1 Introduction

Massive amounts of preserved and shared data are handled by the large-scale data model, which has been made possible by the development and availability of data sources. This makes dispersed research necessary for handling, managing, and processing large amounts of data from many contexts [2]. From the perspective of the application, processed data is shown to offer real-time development solutions. The first stage in obtaining information from various sources, including businesses, homes, the healthcare sector [3], and apps related to road transportation, is large-scale data collection. Processing such data takes more time and involves storage types in alongside equipment architectural support. However, statistical analysis offers the best answers through simplifying lengthy information into understandable visual chunks [1, 4]. Massive raw data evaluation and administration are carried out using advanced techniques to lessen the adversarial impact of delays and excellent hardware support [5].

Because the amount of collected data constantly increases based on activities and immediate time demands, comprehensive large-scale data processing is less possible with traditional shared processing platforms [6]. Computation

Corresponding Author: L. S. Dandugala
Received: January 2023; Revised: January 2024; Accepted: April 2024.
https://doi.org/10.22111/IJFS.2024.44621.7855

becomes complicated due to the increased data transfer caused by various program sizes and customer needs. Large-scale data management also involves managing the retention and retrieval of data [7]-[9]. To overcome the problems in big data analytics, several data management strategies are put forth within reality [10].

A fast and unsupervised ML technique called clustering can be used to find items in a database that belong to the same category [11]. K-Means is the grouping method used more often because of its exceptional quality outcomes and short processing time [13]. However, the amount of iterations in a K-Means repetition tends to boost its runtime, so using the approach known as K-Means for document clustering may result in unanticipated time costs [12]. The need for efficient document grouping methods has led researchers to search for feasible tweaks for well-proven algorithms like K-Means [14]-[16].

Apache created and maintained Hadoop, an open-source platform for the worldwide processing of large datasets [17]. With low-cost technology, Hadoop handles fewer data bytes, which is known as HDFS (Hadoop distributed file system). MapReduce provides a platform for concurrent processing and efficiently leverages the benefits of meta-heuristic algorithms for studying large datasets. Many distributed evolutionary algorithms have lauded the straightforwardness of the MapReduce design for its ability to address a wide range of technologically challenging issues [18]. Consequently, researchers have created simultaneous meta-heuristic algorithms utilizing MapReduce over the last five years.

Big data clustering research is necessary and motivated by the challenges presented by the unparalleled volume, variety, and velocity of data in contemporary contexts. Conventional clustering techniques find it difficult to process large datasets quickly enough to extract meaningful patterns in the era of big data [19]. These dataset's immense size and complexity necessitate the development of novel clustering strategies that can manage high-dimensional data, scale flawlessly, and react to immediate information streams. The growing frequency of noise, outliers, and the requirement for interpretability in clustering results further increases the demand for sophisticated approaches. The need to create scalable, effective, and adaptive algorithms that can find hidden patterns in large datasets is driving studies on big data clustering [20]. The ultimate objective is to enable decision-makers with knowledge obtained from the complex relationships contained in big data, promoting well-informed decision-making in various fields, including finance, telecommunications, healthcare, and more. Therefore, we proposed a novel fuzzy-based approach and ensemble approach to enhance the clustering performances. Our proposed approach gains less memory usage, low computational time, and improved performance compared to existing approaches.

1.1 Novel contribution

Existing research faces high computational complexity resulting from the enormous amount of data, which is one of the drawbacks of big data clustering, frequently impeding real-time processing. Achieving the best clustering results is further hampered by the difficulties in managing high-dimensional and diverse data, as well as the higher reactivity to parameter circumstances. We proposed a novel ensemble approach to tackle the existing limitations, which enhances efficiency.

- To boost the performance of clustering, we presented a novel Fuzzy based Improved Multiview Fuzzy C-Means clustering approach that calculates the separation among the current instance and the Cluster head (CH).

- We ensemble MobileNet V2 and three layer stacked BiLSTM termed as MVSBiLSTM approach to boost the mechanism of clustering.

- We created a map/reduced computing for the proposed strategies implemented in a Hadoop environment.

- We evaluate the performance of the suggested approach on three benchmark datasets. The outcomes imply that the suggested approach performs better in selecting the cluster.

This is the format for the rest of the paper. The literature is reviewed in Part 2. Part 3 covered the problem description and strategy in detail. The outcomes and debates are highlighted in Part 4. We conclude the work in Part 5. Abbreviation and symbols are shown in Table 1.

Table 1: Abbreviation and explanation

Abbreviation	**Explanation**
BDA	Big Data Analysis
IMV-FCA	Improved Multi-View Fuzzy C-Means Algorithm
MVSBiLSTM	Mobilenet V2 Model And Three Layered Stacked Bidirectional LSTM
ML	Machine Learning
HDFS	Hadoop Distributed File System
CH	Cluster Head
FDM	Fuzzy-Optimized Data Management
GWO	Grey Wolf Optimizer
EGWO	Enhanced Grey Wolf Optimizer
YOLOv4	You Only Look Once Version 4
SVM	Support Vector Machine
BS	Base Station
CNN	Convolutional Neural Network
RNNs	Recurrent Neural Ntworks
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
HDFS	Hadoop File System
RAM	Random Access Memory
CPU	Central Processing Unit
GB	Gigabyte
GHz	Gigahertz
CMC	Contraceptive Method Choice
NMI	Normalized Mutual Information
ARI	Adjusted Rand Index

2 Related works

Analysis of big data was the focus of numerous studies. The drawback of big data clustering is its high memory requirements. To address this, researchers propose novel approaches that expand on previously developed clustering algorithms.

In order to recognize and improve the grouping of accumulated knowledge-based semantics and limitations, Manogaran et al. [21] developed fuzzy-optimized data management (FDM). The connections modeled between data points according to the values of attributes are used to classify data dependency. By classifying the qualities taken into consideration according to similarity index boundaries, this approach efficiently processes complex data. The recommended strategy performs better when compared to the current ones.

A MapReduce-based MR-EGWO for big data analysis clustering is described by Tripathi et al. [22]. The proposed approach produced a distinct variation of the GWO, the EGWO, where the binomial crossover is combined with the wolf's foraging strategy, and longer flight durations are added to boost the wolf's ability to pray. The clustering process is then optimized using the suggested modification. The suggested approach is a potent and precise depth for suitable clustering based on the investigation's outcomes. The En-ABC was introduced by Garg et al. [23] for cloud-based multi-class databases. Finally, an effective comparison has been made between the proposed scheme and existing techniques. The recommended strategy produces better performances, according to the experimental findings.

In order to concentrate its efforts, Zhong et al. [24] suggested that BDHDLS learn the unique data distribution of particular invasive attacks that belong to particular families. To comprehend network traffic characteristics and data contained in the payload, BDHDLS uses both behavioral and content aspects. Every deep learning model within the BDHDLS focuses on acquiring knowledge of the distinct data distribution within a single cluster.

Zhu et al. [26] provide a technique for anomaly identification based on traffic patterns. First, a city is divided into discrete groups based on the similarities in activity series dynamics between various geographic areas. Next, k-means clustering is used in each grouped area to find abnormalities. Finally, apply the established method to a real-world data set, such as the aggregate CDR data from Milan, Italy, and use ground truth data to confirm the method's efficacy. A solid method for picture recognition and categorization of large medical data utilizing YOLOv4 and a logistic regression approach was offered by Awad et al. [28]. To increase the effectiveness of these techniques, researchers proposed

Table 2: Overview of prior works

Ref	Approaches	Difficulties	Advantages
Manogaran et al. [21]	fuzzy-optimized data management (FDM)	The evaluation of memory usage was not conducted.	The memory usage of the suggested system has been assessed.
Tripathi et al. [22]	enhanced grey wolf optimizer	Require a large amount of processing time	Our suggested method requires less processing time.
Garg et al. [23]	Ensemble Artificial Bee Colony	Processing effectiveness and efficiency must increase.	For the purpose of assessment, four publicly accessible datasets were utilized.
Zhong et al. [24]	BDHDLS	It is still necessary to increase accuracy.	Our suggested framework shows a notable increase in accuracy.
Zhu et al. [25]	K-Means	At the expense of increased memory usage, deliver greater accuracy	Designed a clustering algorithm that uses the least amount of memory and is accurate.
Awad et al. [26]	Yolo V4	The computational complexity of the presented approach is high	Presented approach gains improved performances
Zhou et al. [27]	GraphX	It takes a large amount of processing time	Map reduce concept is implemented
Li et al. [28]	K-Means	High computational complexity leads to lower performances	Memory usage is less

employing complex parallel k-means preprocessing, a clustering technique that identified structure and pattern in the data. We additionally leveraged the acceleration features of a neural engine processor to further enhance the efficiency and speed of the method. It proved the method's ability to accurately identify medical images and classify substantial amounts of medical data by testing it on several large-scale medical datasets.

In order to enhance the effectiveness of financial fraud detection, a distributed and intelligent Big Data method is suggested by Zhou et al. [27]. Four modules comprise the approach: the typical data feature module, the prediction module, the graph embedding module, and the data pretreatment module. The topological characteristics of every vertex in the network graph are learned and represented into a low-dimensional dense vector utilizing the graph embedding method Node2Vec, which is used with Spark GraphX and Hadoop.

Li et al. [25] introduce a method to find clusters in financial data and maximize the size of the clusters for easy interpretation. Specifically, researchers can direct basic clustering techniques like k-Means to find hyperellipsoidal clusters automatically. The new cluster quality evaluation criterion is free from large-scale computing. Next, created a new solver for an updated support vector data description model. This solver effectively improves the centroids and scopes of the identified clusters, making the clusters tighter and easier to understand using eigenvectors because the data within the clusters is more similar. The trials show that the suggested approach can effectively locate an acceptable number of clusters using ten financial datasets. An overview of related work is shown in Table 2.

2.1 Challenges

The following illustrates the work's challenges:
Due to the data was collected from multiple sources without a standard structure, the big, diverse, and dynamic nature of the method presents its biggest challenge to big data clustering. Big data requires a lot of time to process because it contains enormous volumes of data and many different types of data objects. Most conventional techniques suffer from computational difficulty and take a long time to process. This work established an innovative big data clustering strategy employing the suggested ensemble approach, considering the shortcomings of the currently available big data clustering techniques as motivation. The suggested approach tries to address the difficulties of big data clustering.

3 Methodology

Big data clustering is the process of organizing and interpreting enormous and complex datasets using scalable and effective algorithms. It is essential for revealing hidden frameworks, relationships, and patterns in large data sets, which helps with decision-making and extracting insights.

This article introduces a novel Improved Multiview Fuzzy C-Means approach (IMV-FCA) based on parallel batch clustering. The proposed method for utilizing a divided dataset involves several steps. The input dataset is divided into batches. When clustering is used, each batch is handled as a distinct dataset. The starting centroids are chosen at random. Batches are processed concurrently until the convergence condition is met. The technique lowers the total squared errors for each cluster. Every batch's centroids produce an updated minimal dataset, to which clustering is

once more used to locate the centroids. Utilizing parallel processing and batch-segmenting data prior to clustering reduces computation time. To improve performance and accelerate computation, an ensemble of stacked BiLSTM and MobileNet V2, referred to as MVSBiLSTM, is employed. To reduce algorithmic temporal complexity and boost algorithmic effectiveness, the map-reduce strategy is used.

3.1 Improved multiview fuzzy C-means clustering (IMV-FCM) approach

Utilizing the IMV-FCM strategy is primarily motivated by improving cluster effectiveness, robustness, and accuracy in big data analytics. Using multiple views, the method can capture different aspects of the data, leading to more insightful and accurate clustering results. This approach may also help with better management of highly dimensional and heterogeneous databases, which are common in big data applications. The ultimate goal is to provide a more effective and scalable way to organize large and intricate datasets to extract important information and insights from big data sources. Located at what is thought to be the center of the circled network area is the base station (BS). A specific number of rings separate the network, every with a predetermined diameter. The offline part and the grouping phase are the two main components of the proposed clustering method. We then use a fuzzy approach for each clustering stage to analyze the proper CH.

Clustering algorithm

These characteristics depend on various aspects, including the composition of big data. Cluster analysis divides the data points with similar properties and attempts to produce a fair analysis based on the sample's features. Taxonomy serves as the inspiration for the cluster analysis algorithm. The need for analysis is growing as big data includes more and more information. Multivariate analysis has been incorporated with taxonomy to create cluster analysis. Cluster analysis aims to gather information for analysis when the information points are similar. One method for examining the connection between sample information points is cluster analysis. The clustering results offer a crucial foundation for data processing by highlighting the similarities and differences within the sample data.

The grayscale group of pixels is referred to as the clustering object. A crucial step is to utilize a similarity measure to gauge how similar two data points are. The similarity metric of Euclidean distance is frequently employed. The clustering method selects and groups data points extremely near to one another using similarity measurements. There is also a feedback process because the outcome of clustering influences the choice of following objects and the similarity metric. Evaluating the clustering results is the final step in the clustering process. Evaluating the clustering findings will be as important as the procedures before. For instance, in a cluster analysis method where the threshold value is set, the analysis result must be evaluated to determine the threshold value. The choice of the appropriate criterion function in the cluster analysis method using the function as the judgment or evaluation of the analysis result might guide the criterion.

The analysis of clusters is started at the initial stage by assigning the nodes that belong to the essential levels of energy. The suggested protocol is based on network diversity, where nodes use different amounts of energy to fulfill their responsibilities. Normal and power nodes are the categories into which the proposed scheme splits nodes. Power nodes have far less initial energy than regular nodes. The energy level is represented by the following expressions (1) and (2):

$$\text{Norma lno des}E_{normal} = E_0, \tag{1}$$

$$PowernodesE_{power} = E_0 * (1+?), \tag{2}$$

The overall energy allocation throughout the network can be written like this:

$$n \cdot E(1-m) + n \cdot m \cdot E(1+?) = n \cdot E(1+? \cdot m), \tag{3}$$

The number of nodes is represented as n is shown in equation 3, the percentage of power nodes is m, and the sum of the energy is denoted by h. At protocol startup, the quantity of CH utilized during each transmission phase is calculated.

Multiview clustering algorithm

Traditional fuzzy clustering cannot produce the best outcomes when handling multiview data. Many methods related to multiview have been proposed based on this need. Multitask, collaborative, and clustering clustering algorithms are

some of the multi-view-related clustering techniques now in use. While there are other multiview clustering techniques, CoFKM is the most well-known. We will then quickly go over the algorithm's concept and its phases of implementation.

The function expression for CoFKM is provided by:

$$J_{CoFKM} = \sum_{\nu=1}^{V}\sum_{i=1}^{C}\sum_{j=1}^{N}\tilde{u}_{i,j,\nu,\eta}||x_{j,\nu} - Z_{i,\nu}||^2, \tag{4}$$

$$\tilde{u}_{i,j,\nu,\eta} = (1-\eta)u_{i,j,\nu}^{m} + \frac{\eta}{\nu-1}\sum_{\nu'=1,\nu'\neq\nu}^{V} u_{i,j,\nu'}. \tag{5}$$

$$\begin{cases} u_{i,j,\nu} \in [0,1], 1 \le i \le C,\ 1 \le j \le N \\ \sum_{i=1}^{c} u_{i,j,\nu} = 1,\ 1 \le v \le V \end{cases}$$

The membership degree and clustering center calculation formulas are produced by applying the Lagrangian extremum solution approach, as per Eq. (5). The membership matrix and cluster center computation formulas for the ν^{th} view are provided by equations (4) and (5).

$$z_{i,\nu} = \frac{\sum_{j=1}^{N}\tilde{u}_{i,j,\nu,\eta}x_{i,\nu}}{\sum_{j=1}^{N}\tilde{u}_{i,j,\nu,\eta}},\ i = 1,2,\cdots,C, \tag{6}$$

$$u_{i,j,\nu} = \frac{1}{\sum_{?=1}^{C}\left[\dfrac{(1-\eta)d_{i,j,\nu}^2 + \dfrac{\eta}{V-1}\sum_{\nu'=1,\nu'\neq\nu}^{V} d_{i,j,\nu'}^2}{(1-\eta)d_{?,j,\nu}^2 + \dfrac{\eta}{V-1}\sum_{\nu'=1,\nu'\neq\nu}^{V} d_{?,j,\nu'}^2}\right]}, \tag{7}$$

The division matrix for any view is computed using Equations (6) and (7). The geometric mean ensemble approach is used to get the final division result. The integrated method's expression is shown in expression (8):

$$\hat{U} = \sqrt[\nu]{\prod_{\nu=1}^{V} U_\nu}. \tag{8}$$

IMV-FCM

The CoFKM presented in the "Multiview Clustering Algorithm" section gives each view the same weight, indicating that it thinks each view's data contributes equally to the grouping. This doesn't match the actual circumstances. Each view data's contribution to the clustering for MVD in actual production and living contexts must differ. The overall performance of the clustering process can be enhanced by assigning a high weight to the angle of view data that contributes much and a low weight to the angle of view data that contributes less. As a result, this study uses a multiview fuzzy clustering algorithm with the capacity to modify viewing angles adaptively. Its goal is

$$J = \sum_{\nu=1}^{V}\sum_{i=1}^{C}\sum_{j=1}^{N}\sum_{t=1}^{V} w_{v,t}u_{i,j,t}^{m}d_{i,j,\nu}^2 + \gamma\sum_{\nu=1}^{V}\sum_{t=1}^{V} w_{\nu,t}\log(w_{v,t}), \tag{9}$$

$$\sum_{i=1}^{C} u_{i,j,\nu} = 1, u_{i,j,\nu} \in [0,1]. \tag{10}$$
$$1 \le j \le C, 1 \le j \le N, 1 \le v \le V$$

$$\sum_{t=1}^{V} w_{\nu,t} = 1,\ w_{\nu,t} \in [0,1],\ 1 \le \nu \le V. \tag{11}$$

In the i^{th} perspective, the division matrix is represented by $d^2_{i,j,\nu} = ||x_{j,\nu} - z_{i,\nu}||^2$ and $U_V = [U_{ij,v}]$. $\tilde{u}_{i,j,\nu} = \sum_{t=1}^{V} w_{\nu,t} u^m_{i,j,t}$ is the division fusion item, which in the ν^{th} view clustering task realizes the view fusion of various views shown in equations (9), (10) and (11). At this point, the weight wv, t, which indicates the significance of the t^{th} view in the ν^{th} view cluster, reflects the relevance of each perspective. The weight matrix for every view is W. Equation (11)'s extreme value is solved using the Lagrange multiplier method, and the following expressions for each variable are obtained:

$$z_{i,\nu} = \frac{\sum_{j=1}^{N} \sum_{t=1}^{V} w_{\nu,t} u^m_{i,j,t} x_{j,\nu}}{\sum_{j=1}^{N} \sum_{t=1}^{V} w_{\nu,t} u^m_{i,j,t}}, \tag{12}$$

$$u_{i,j,t} = \frac{1}{\sum_{?=1}^{C} \left[\frac{\sum_{\nu=1}^{V} w_{\nu,t} d^2_{i,j,\nu}}{\sum_{\nu=1}^{V} w_{\nu,t} d^2_{?,j,\nu}} \right]^{\frac{1}{(m-1)}}}, \tag{13}$$

$$W_{\nu,t} = \frac{exp\left[\frac{-\sum_{i=1}^{C} \sum_{j=1}^{N} u^m_{i,j,t} d^2_{i,j,\nu}}{\gamma} \right]}{\sum_{g=1}^{V} exp\left[\left[\frac{-\sum_{i=1}^{C} \sum_{j=1}^{N} u^m_{i,j,g} d^2_{i,j,\nu}}{\gamma} \right] \right]} \tag{14}$$

To acquire the final partition matrix, apply the subsequent ensemble method:

$$\bar{U} = \max \left(\sum_{t=1}^{V} \left(\frac{\sum_{\nu=1}^{V} w_{\nu,t}}{\sum_{t=1}^{V} \sum_{\nu=1}^{V} w_{\nu,t}} U_t \right) \right) \tag{15}$$

The final global membership matrix is created using the fusion approach of Equation 15, with a matching weight w assigned to every membership matrix.

3.1.1 Energy node ratio

For cluster-based frameworks, a significant obstacle is energy. Energy is a crucial component of the CH process. The cluster head selection process in the suggested method uses the node's starting and ending energy ratios. The CH selection process is energy-based due to the energy ratio utilized in the fitting function of the suggested strategy. The CH competition is more likely to be won by the energy level with the highest probability. The following are the energy parameters for the function:

$$f1 = \sum_{i=1}^{N} \frac{E_0}{E_0 - S(i) \cdot E}, \tag{16}$$

Reducing all energy expenditures aims to minimize node energy ratios in expression (16). A significant amount of energy remains in the node, which lowers the energy ratio's value; for the CH procedure, with the greatest residual energy, nodes were chosen by the fitness function, extending the network's lifespan and efficiency.

3.1.2 Degree of the node

The neighboring nodes will facilitate the formation of a cluster if the node with the greatest number of neighbors is designated as the CH. At a low energy cost, neighboring nodes can connect to their nearby CH. By selecting the node with the most neighbors as the Cluster head using the node degree, the function saves the cluster member energy required to transport data to the Cluster head. This parameter aims to reduce the number of nodes in the CH nomination pool.

$$f3 = \frac{1}{\sum_{i=1}^{N} \text{node degree } (i)}. \tag{17}$$

Number of nodes is represented as N in eq (17).

3.2 MobileNet V2

A prevalent model for enhancing cluster efficiency is MobileNet V2, which is based on CNN. Utilizing the design has the main benefit of requiring a lot less computational work than the CNN approach, which makes it appropriate for usage on low-processing PCs and mobile devices. Deep neural networks are used extensively in many applications, including computer vision, with particular architectures such as convolutional neural networks (CNNs) and MobileNet V2. The purpose of CNNs, like MobileNet V2, is to extract hierarchical representations from incoming data automatically. Deep neural networks perform well in contexts with limited resources, as shown by MobileNet V2, a CNN variation designed for mobile and edge devices. Its architecture includes depthwise separable convolutions, which reduce computing costs without compromising accuracy. This qualifies MobileNet V2 for applications that run in real time on devices with limited processing resources. MobileNet V2 is an excellent choice for big data analysis because it integrates global input through fully connected layers, gathers local patterns using convolutional layers, and pools spatial downsampling through pooling layers. The intricate structure and design of MobileNet V2 show how deep neural networks can extract minute details and relationships from complex, large-scale datasets. A class of designs known as "deep neural networks" includes recurrent neural networks (RNNs) for sequential data and other specialized models beyond MobileNet V2. These networks' depth and complexity allow them to operate exceptionally well at tasks demanding a deep understanding of input data patterns.

A straightforward convolution layer-based architecture called MobileNet can be employed to distinguish among finer that depend on regulated characteristics that effectively alternate between latency and accuracy of the parameter. One benefit of the MobileNet technique is that it lessens network traffic. The degree of difficulty of 1×1 refers to point-wise complexity. The identical parameters and the resolution multiplier are used to reduce the dimension of the image being used and the internal definition of each layer.

The following metrics can be used to assess the amount of c, which represents the overall execution cost for the central layers of abstract design as shown in formula (18):

$$C_e = F_s \cdot F_s \cdot \omega \cdot \alpha F_m \cdot \alpha F_m + \omega \cdot \rho \cdot \alpha F_m. \tag{18}$$

For the statistical evaluation in clustering, the value of the multiplier is believed to be between 1 and n, although it varies based on the situation. Presumably, 1 is the value of the adjustable resolution multiplier.

Within the established structure, a parameter d, which is approximately estimated by the formula (19), is included and limits the depth-wise convolutions.

$$d = \frac{F_s \cdot F_s \cdot \omega \cdot \alpha F_m \cdot \alpha F_m + \omega \cdot \rho \cdot \alpha F_m \cdot \alpha F_m}{F_s \cdot F_s \cdot \omega \cdot \rho \cdot F_m \cdot F_m}. \tag{19}$$

These values should always be greater than 32. The third value indicates that it has three input channels. In the suggested architecture, each of the 32 filters has a size of $3 \times 3 \times 3 \times 32$. The model aims to outperform the traditional convolutional model in simplicity and speed.

3.2.1 Computational model

The computation strategy, which is costly and time-consuming, pre-trains the variables employing CNN as the initial module. In addition, we generate the optimized variant to minimize computation time and effort while maintaining the parameters.

$$?id_\text{layer}_{l_1 \cdots l_p} = enc(\psi) \left(\sum_{k_1 \cdots k_p}^{k_1 \cdots k_p} d^{(1)}_{l_1 \cdots l_p} + Y^{(1)}_{ak_1 \cdots k_?} Z_{k_1 \cdots k_?} \right), \tag{20}$$

$$\text{Out_layer}_{k_1 \cdots k_?} = dec(\psi) \left(\sum_{l_1 \cdots l_0}^{L_1 \cdots L_p} d^{(1)}_{k_1 \cdots k_?} + Y^{(1)}_{\beta l_1 \cdots l_0} ?\ id_\text{layer}_{l_1 \cdots l_0} \right). \tag{21}$$

The total number of dimensions is denoted by $K1$, the hidden layer is represented by $L1$, the encoder is identified by enc, and dec identifies the decoder. The function of sigmoid is also utilized in both the decoding and encoding layers

shown in eqn (20) and (21). The rebuilding objective is described by the formula (**??**). The goal of reconstruction is

$$L_{vencdec}(\psi) = \left[\frac{1}{0} \sum_{m=1}^{0} \left(\sum_{s=1}^{k_1 x \cdots x k_0} \sum_{l_1=1}^{L_1} \cdots \sum_{l_0=1}^{L_p} \left(Y^{(2)}_{sl_1 \cdots l_0} \right) \right)^2 + 0.5(\text{out_layer}_m - z_m)^{\nu} I((\text{out_layer}_m - Z_m)) \right.$$
$$\left. + 0.5\xi \left(\sum_{r=1}^{L_1 x \cdots x L_n} \sum_{k_1=1}^{k_1} \cdots \sum_{ip=1}^{k_0} Y^{(1)}_{rk_1 \cdots k_0} \right)^2 . \right. \tag{22}$$

Moreover, the parameter is trained using backpropagation.

3.2.2 Three-Layer stacked bidirectional LSTM

We present the three-layer stacked bidirectional long short-term memory (LSTM) approach to enhance the cluster performances. To more effectively manage sequential and spatial features, capture dependence over time, and retrieve significant trends from the complicated patterns inherent in large datasets, stacked bidirectional LSTMs are used in big data analysis. This methodology enhances the precision and broadness of analyses, enabling significant discoveries and insights from large datasets. Utilizing n-grams as data representations, we initially developed a weighted embedding of words approach. The foundation of this plan is a term-weighted word embedding framework based on inverse gravity moments. The weighted word-embedding approach that was developed has been fed into the stacked BiLSTM framework. The contextual information was further obtained for displaying textual information within this framework after analyzing big data. Following this, a softmax layer combined the outputs.

To improve the efficiency, we have created a three-layer stacked bidirectional LSTM framework, which depends on qualitative research with varying layers. It has been pointed out that bidirectional architectures can outperform unidirectional ones. Stacking systems can also achieve higher accuracy in prediction since they obtain richer contextual data about past and future data sequences. The stacked LSTM frameworks with more upper layers can provide more contextual details regarding the data, leading to an enhanced prediction model.

The input data was transmitted to the layers that are hidden forward for a particular time period T to learn about earlier phases. The sequence of inputs was additionally fed to the layers hidden in the opposite direction so they can learn about the following stages. Units of layers that are concealed in higher hierarchies of the method recognize particular contextual details by utilizing results from the lower levels of the setup.

The detected concealed state information (a_t) for the initial forward layer is given by expression 23-28:

$$i_t^{(a)} = \sigma\left(W_i^{(a)} a_{t-1} + U_i^{(a)} x_t + b_i^{(a)}\right), \tag{23}$$
$$f_t^{(a)} = \sigma\left(W_f^{(a)} a_{t-1} + U_f^{(a)} x_t + b_f^{(a)}\right), \tag{24}$$
$$o_t^{(a)} = \sigma\left(W_o^{(a)} a_{t-1} + U_o^{(a)} x_t + b_o^{(a)}\right), \tag{25}$$
$$u_t^{(a)} = \tanh\left(W_u^{(a)} a_{t-1} + U_u^{(a)} x_t + b_u^{(a)}\right), \tag{26}$$
$$C_t^{(a)} = i_t^{(a) o u_t(a)} + f_t^{(a) o C_{t-1}(a)}, \tag{27}$$
$$a_t = o_t^{(a) o} \tanh\left(C_t^{(a)}\right). \tag{28}$$

For the following forward layer, the determined hidden data for the state (b_t) is given by expressions 29-34.

$$i_t^{(b)} = \sigma\left(W_i^{(b)} b_{t-1} + U_i^{(b)} x_t + b_i^{(b)}\right), \tag{29}$$
$$f_t^{(b)} = \sigma\left(W_f^{(b)} b_{t-1} + U_f^{(b)} x_t + b_f^{(b)}\right), \tag{30}$$
$$o_t^{(b)} = \sigma\left(W_o^{(b)} b_{t-1} + U_o^{(b)} x_t + b_o^{(b)}\right), \tag{31}$$
$$u_t^{(b)} = \tanh\left(W_u^{(b)} b_{t-1} + U_u^{(b)} x_t + b_u^{(b)}\right), \tag{32}$$
$$C_t^{(b)} = i_t^{(b) o u_t(b)} + f_t^{(b) o C_{t-1}(b)}, \tag{33}$$
$$b_t = o_t^{(b) o} \tanh\left(C_t^{(b)}\right). \tag{34}$$

The recognized hidden data state (c_t) for the 3rd forwarding layer is given by the expression 35-40.

$$i_t^{(c)} = \sigma\left(W_i^{(c)} c_{t-1} + U_i^{(c)} x_t + b_i^{(c)}\right), \tag{35}$$

$$f_t^{(c)} = \sigma\left(W_f^{(c)} c_{t-1} + U_f^{(c)} x_t + b_f^{(c)}\right), \tag{36}$$

$$o_t^{(c)} = \sigma\left(W_o^{(c)} c_{t-1} + U_o^{(c)} x_t + b_o^{(c)}\right), \tag{37}$$

$$u_t^{(c)} = \tanh\left(W_u^{(c)} c_{t-1} + U_u^{(c)} x_t + b_u^{(c)}\right), \tag{38}$$

$$C_t^{(c)} = i_t^{(c)ou_2(c)} + f_t^{(c)oC_{t-1}(c)}, \tag{39}$$

$$c_t = o_t^{(c)o} \tanh\left(C_t^{(c)}\right). \tag{40}$$

Equation 43 has revealed the hidden state data (d_t) for the first backward layer in the expression 41-46.

$$i_t^{(d)} = \sigma\left(W_i^{(d)} d_{t-1} + U_i^{(d)} x_t + b_i^{(d)}\right), \tag{41}$$

$$f_t^{(d)} = \sigma\left(W_f^{(d)} d_{t-1} + U_f^{(d)} x_t + b_f^{(d)}\right), \tag{42}$$

$$o_t^{(d)} = \sigma\left(W_o^{(d)} d_{t-1} + U_o^{(d)} x_t + b_o^{(d)}\right), \tag{43}$$

$$u_t^{(d)} = \tanh\left(W_u^{(d)} d_{t-1} + U_u^{(d)} x_t + b_u^{(d)}\right), \tag{44}$$

$$C_t^{(d)} = i_t^{(d)ou_2(d)} + f_t^{(d)oC_{t-1}(d)}, \tag{45}$$

$$d_t = o_t^{(d)o} \tanh\left(C_t^{(d)}\right). \tag{46}$$

Equation 49 has determined the hidden state data (e_t) for the second backward layer that is shown in 47-52:

$$i_t^{(e)} = \sigma\left(W_i^{(e)} e_{t-1} + U_i^{(e)} x_t + b_i^{(e)}\right), \tag{47}$$

$$f_t^{(e)} = \sigma\left(W_f^{(e)} e_{t-1} + U_f^{(e)} x_t + b_f^{(e)}\right), \tag{48}$$

$$o_t^{(e)} = \sigma\left(W_o^{(e)} e_{t-1} + U_o^{(e)} x_t + b_o^{(e)}\right), \tag{49}$$

$$u_t^{(e)} = \tanh\left(W_u^{(e)} e_{t-1} + U_u^{(e)} x_t + b_u^{(e)}\right), \tag{50}$$

$$C_t^{(e)} = i_t^{(e)ou_2(e)} + f_t^{(e)oC_{t-1}(e)}, \tag{51}$$

$$e_t = o_t^{(e)o} \tanh\left(C_t^{(e)}\right). \tag{52}$$

Equation 55 provides the identified hidden state data (g_t) for the third backward layer, which is shown in (53-58).

$$i_t^{(g)} = \sigma\left(W_i^{(g)} g_{t-1} + U_i^{(g)} x_t + b_i^{(g)}\right), \tag{53}$$

$$f_t^{(g)} = \sigma\left(W_f^{(g)} g_{t-1} + U_f^{(g)} x_t + b_f^{(g)}\right), \tag{54}$$

$$o_t^{(g)} = \sigma\left(W_o^{(g)} g_{t-1} + U_o^{(g)} x_t + b_o^{(g)}\right), \tag{55}$$

$$u_t^{(g)} = \tanh\left(W_u^{(g)} g_{t-1} + U_u^{(g)} x_t + b_u^{(g)}\right), \tag{56}$$

$$C_t^{(g)} = i_t^{(g)ou_2(g)} + f_t^{(g)oC_{t-1}(g)}, \tag{57}$$

$$g_t = o_t^{(g)o} \tanh\left(C_t^{(g)}\right). \tag{58}$$

Equation 59 illustrates how the outcome has been acquired for a specific time step t by concatenating the hidden state data (g_t) for the third backward layer and the hidden state data (c_t) for the third forward layer:

$$o_t = U^{(O)} c_t + W^{(O)} g_t + b^{(O)}. \tag{59}$$

After receiving the outcome for the last time phase, the softmax layer was employed to determine the last class label for the framework, as shown by the expression 61, while y 0 is the expected class label for a particular instance:

$$p(y(X)) = softmax\left(W^{(s)}O_K + b^{(s)}\right). \tag{60}$$

$$y' = argmax_y\ p(y(X)). \tag{61}$$

3.3 Map reduce

MapReduce is a popular processing platform and computing paradigm utilized in huge data environments for distributed and simultaneous processing. It consists of two primary stages, Map and Reduce, which can be tailored to different big data clustering workflow phases. The Map phase involves breaking up the data into smaller pieces and having mappers do things like cleaning, preprocessing, and extracting features. This stage allows for the efficient management of high-dimensional data, which is typical in big data scenarios. In order to prepare the data for the ensuing Reduce phase, the Shuffle and Sort step arranges mapper results according to keys.

Reducer's procedure grouped data and produced their final clustering outcomes during the Reduce stage by applying algorithms for clustering. During this stage, outcomes from various mappers are combined, which improves the clustering procedure's scalability and effectiveness. It is possible to enhance or modify the clustering results by adding post-processing phases. Although MapReduce isn't a clustering algorithm per se, its distributed and parallel architecture makes it helpful in managing big datasets and speeding up the processing of different tasks related to clustering in big data circumstances. Figure 1 depicts these procedures.

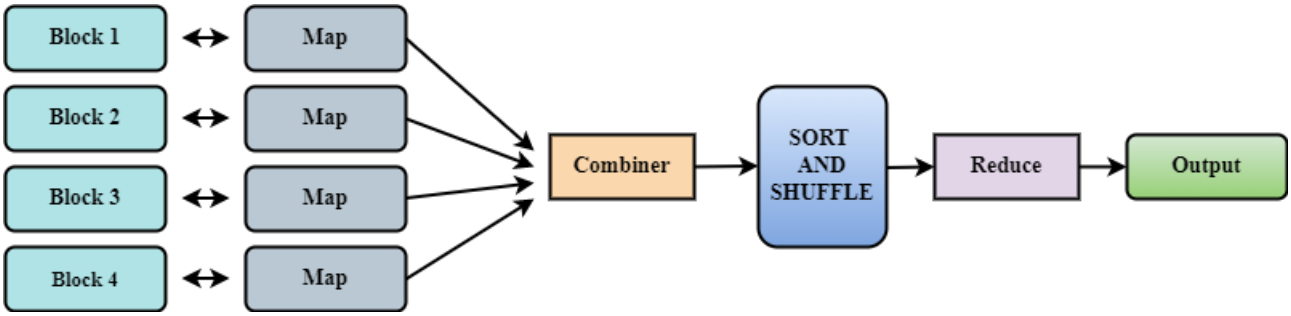


Figure 1: Map and reduce task

Each Map phase's input information is a subset of the data entered. The input data is gathered, the function known as Map is used, and the outcome of this stage is sorted and combined by the Map tasks. The outcome is reversed, and the transference time ought to be raised rather than lowered if there are a great number of nodes. Every component has weights of M1, M2, R1, R2, and R3 in its mapping stage, and every component has an established pace of progress throughout the reduction of weight stage.

4 Result and discussions

The analytical results acquired using the proposed method are shown in this section, which is based on assessment criteria such as kappa, f1-score, sensitivity, accuracy, and specificity.

4.1 Environment setup

A group of commodity PCs serves as the mechanism's testing setup, and HDFS is the repository. Consequently, Hadoop 2.51 has been deployed on this system, and Ubuntu Linux with kernel 2.6.24 is operating on every server within the cluster. This is a multi-node cluster, so the node from master has 16 GB of RAM and is separated from the other workstations, which are categorized as slave nodes and have 16/4 GB of RAM. The master node processor features a four-core virtualized CPU and operates at 2.4 GHz. To showcase how map-reduce programming can save response times.

4.2 Dataset description

We evaluated and simulated our proposed approach performance over three datasets. The explanation of the datasets is given below:

4.2.1 Mfeat data set

The Mfeat dataset, which consists of 2000 numerical objects categorized into ten groups (0–9), is a written-by-hand mathematical dataset generated from an assortment of Dutch utility maps. There are 200 items in each class. Six views, totaling 649 characteristics, are used to describe each item.

4.2.2 Spambase data set

The 4601 items in the Spambase data set are divided into two categories: non-spam emails gathered from research and personal emails (spam, non-spam), and spam messages gathered from postmasters and people who witnessed field spam. Each item has three views, each with fifty-seven features.

4.2.3 CMC dataset

The Contraceptive Method Choice (CMC) database is the second dataset. This database includes information on married women's preferred method of contraception, as well as demographic and economic data. To meet the binary categorization criteria, the characteristic of the contraceptive approach was divided into three divisions: those who use contraception, those who do not, and others. The CMC consists of three categories, 9 characteristics, and 1473 points of data.

4.3 Evaluation metrics

The efficacy of the strategies for clustering can be investigated by comparing the cluster that is generated to the actual structure using external requirements as the true subdivisions of the three data sets employed in the suggested investigations.

4.3.1 Accuracy

To be deemed accurate, data must accurately depict the facts and come from trustworthy sources. Equation 62 is used to estimate the accuracy.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{62}$$

4.3.2 Sensitivity

Sensitivity is also calculated as the proportion of every pair of points found and is determined on a collection of data points. Cluster efficiency and recall are related as well. Equation 63 is used to estimate the sensitivity.

$$Sensitivity = \frac{TP}{TP + FN} \tag{63}$$

4.3.3 Specificity

A measure of specificity is the proportion of pairings of points of data that belong to the identical cluster. It varies in direct proportion to how well the clusters form. Equation 64 is used to estimate the specificity.

$$Specificity = \frac{TP}{TP + Fp} \tag{64}$$

4.3.4 F1-Score

The calculation of the F1 score is intimately linked to recall and precision; Equation 65 is used to estimate the F1 score.

$$F1 - Score = 2 * \left(\frac{precison * recall}{precison + recall} \right) \tag{65}$$

4.4 Normalized mutual information (NMI)

The mutual reliance between two variables is measured by the MNI. One on the NMI scale denotes perfect correlation, while zero indicates no information sharing. Equation 66 is used to estimate the NMI.

$$NM1 = (C(D) + C(A))(C(D, A))^{-1}. \tag{66}$$

4.4.1 Adjusted rand index (ARI)

Comparing two distinct data clustering, the ARI is only a similarity metric. The values of the Rand Index vary 0 to 1, where 0 indicates that there are always two different data clustering's and 1 indicates that there is only one data clustering. An approach with a greater ARI value is considered more efficient. Equations 67 and 68 are used to estimate the ARI.

$$verageRandindex = (rand_index - truenegative)/(max(Rand_ind) - E(rand_ind))^{-1}. \tag{67}$$

4.4.2 Accuracy

$$Clusteringaccuracy = p\left(\sum_{k=1}^{p} 1(A_k = \max(d_k))\right)^{-1}. \tag{68}$$

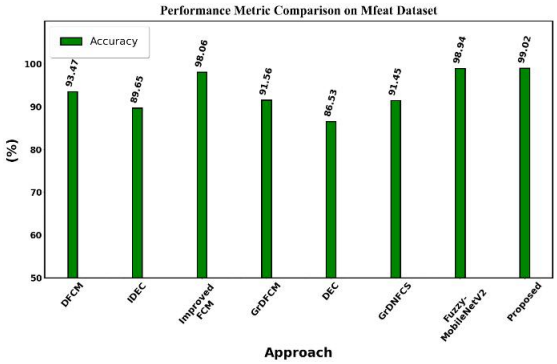
Meanwhile, the clustering task is represented by dk.

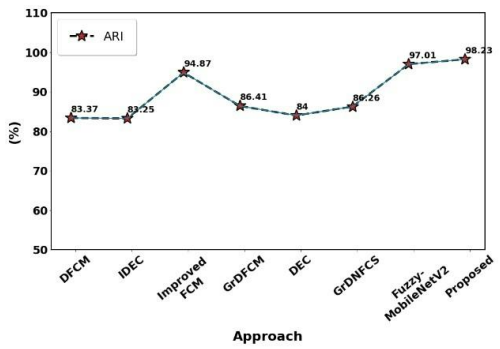
4.5 Performance differentiation of Mfeat dataset

We evaluate the efficacy of the Mfeat dataset in the following section. Table 3 and Figure 2 compare the Mfeat dataset effectiveness.

Table 3: Contrasting the Proposed approach with prior method using dataset 1

Method	Accuracy (%)	ARI (%)	NMI (%)
DFCM	93.47	83.37	88.27
IDEC	89.65	83.25	86.38
Improved FCM	98.06	94.87	94.72
GrDFCM	91.56	86.41	80.49
DEC	86.53	84	84.29
GrDNFCS	91.45	86.26	93.77
Fuzzy-MobileNetV2	98.94	97.01	98.69
Proposed	99.02	98.23	98.97





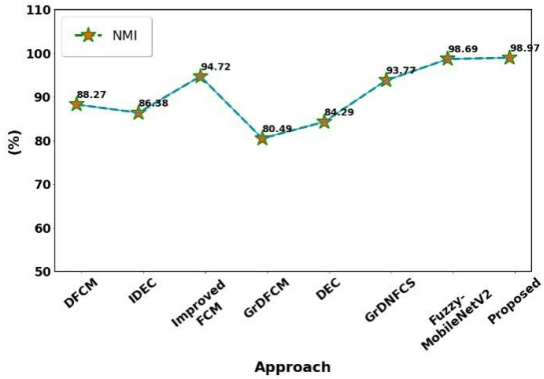


Figure 2: Differentiation of Dataset 1

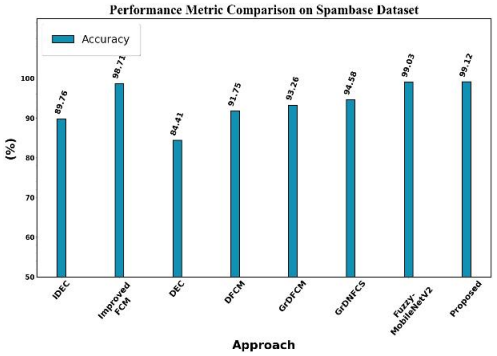
The existing approaches like DFCM, IDEC, Improved FCM, GrDFCM, DEC, GrDNFCS, and Fuzzy-mobileNetV2 are employed to differentiate with the proposed approach. While different from the proposed approach, the presented approach yields superior performance in terms of NMI, Accuracy, and ARI. Our proposed approach achieves 99.02% accuracy, 98.23% of ARI, and 98.97% of NMI. The second-best result presented in the Fuzzy-MobileNet approach is 98.94% accuracy, 97.01% ARI, and 98.69% NMI. The worst performance is present in the DEC approach.

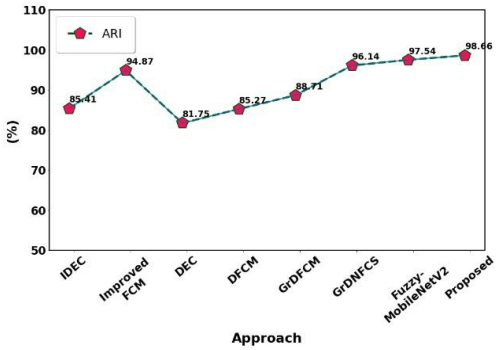
4.6 Evaluation of the spambase dataset

In this section, we simulate the performance of proposed and existing approaches using the spambase dataset. Table 4 and Figure 3 represent the differentiation of proposed with existing approaches using dataset 2.

Table 4: Contrasting Proposed approach with prior method using dataset 2

Approach	Accuracy (%)	ARI (%)	NMI (%)
IDEC	89.76	85.41	84.38
Improved FCM	98.71	94.87	96.44
DEC	84.41	81.75	86.05
DFCM	91.75	85.27	86.54
GrDFCM	93.26	88.71	91.23
GrDNFCS	94.58	96.14	93.56
Fuzzy-MobileNetV2	99.03	97.54	98.47
Proposed	99.12	98.66	99.03





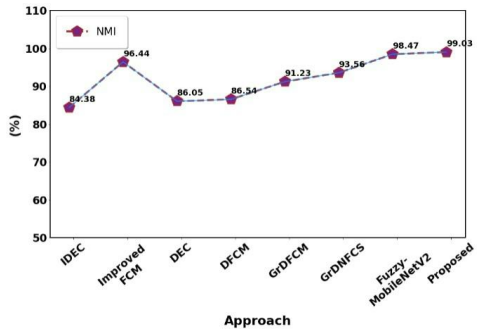


Figure 3: Differentiation of dataset 2

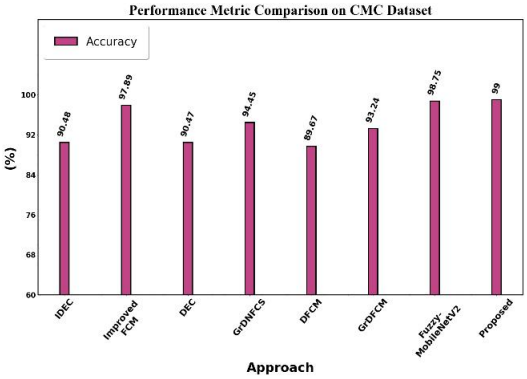
To distinguish using the suggested technique, the following approaches are used: DFCM, IDEC, Improved FCM, GrDFCM, DEC, GrDNFCS, and Fuzzy-mobileNetV2. Although distinct from the suggested strategy, the given approach performs better regarding ARI, NMI, and accuracy. The proposed approach shows that compared with prior approaches, the suggested approach gains a better solution, which is 99.12% accuracy, 98.66% ARI, and 99.03% NMI.

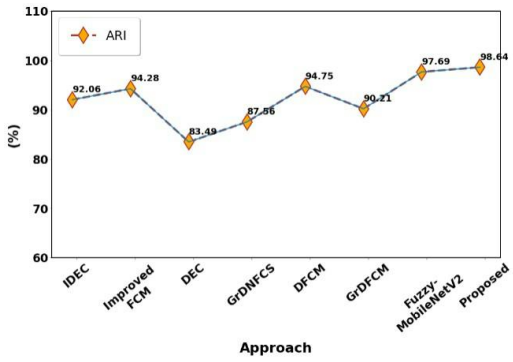
4.7 Evaluation of the CMC dataset

In this section, we analyzed the performance of the proposed with existing approaches using dataset 3. Table 5 compares the proposed approach with existing approach using dataset 3.

Table 5: Contrasting the Proposed approach with prior method using dataset 3

Approach	Accuracy (%)	ARI (%)	NMI (%)
IDEC	90.48	92.06	84.56
Improved FCM	97.89	94.28	96.14
DEC	90.47	83.49	85.39
DFCM	89.67	94.75	86.54
GrDFCM	94.45	87.56	93.21
GrDNFCS	93.24	90.21	90.43
Fuzzy-MobileNetV2	98.75	97.69	98.09
Proposed	99	98.64	98.84





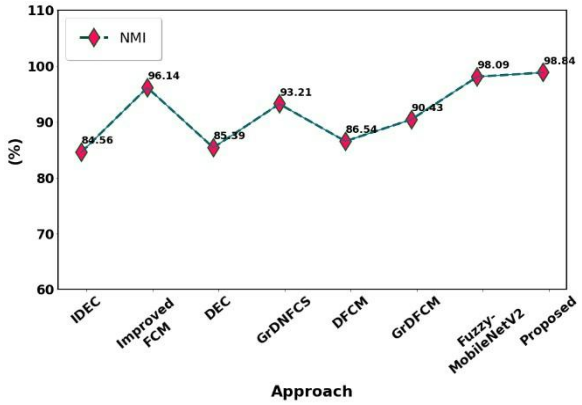


Figure 4: Differentiation of dataset 3

Differentiation of dataset 3 is shown in figure 4. Similarly here also comparing with existing approaches proposed approach achieves greater performances.

4.8 Overall evaluation

In this section, we analyzed the performance of the overall proposal. To simulate and evaluate the performance of the proposed approach, we utilized three datasets.

Table 6: Comparison analysis of proposed dataset

Dataset	F1-Score (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Kappa (%)
Meat	98.34	98.74	97.84	99.02	97.83
CMC	97.57	97.87	98.13	99	96.04
Spambase	96.64	98.88	95.24	99.12	95.61

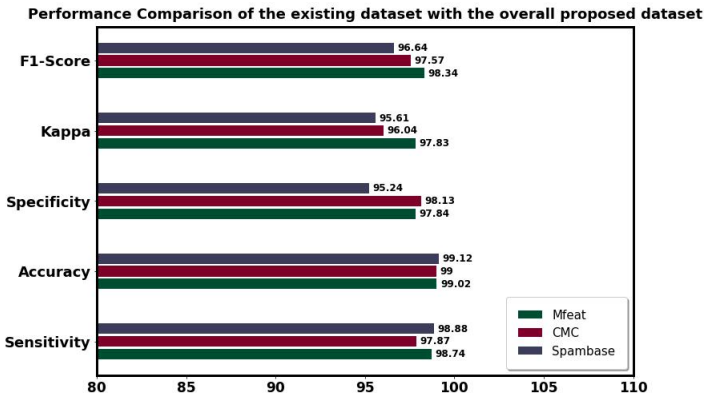


Figure 5: Comparing the suggested dataset's efficiency to the current one

Table 6 and Figure 5 show the comparison of suggested dataset efficiency over existing approaches.

Table 7: Differentiation over existing

Methods	**Computational time**	**Accuracy** (%)
MKS-MRF	6.5	82.43
KFCM	6	89.52
K-Means	8	79.24
FCM based MRF	6	90.60
Sparse FCM	5	93.54
Fuzzy-MobileNetV2	4.3	98.75
FCM	7	79.73
Proposed	3.7	99.06

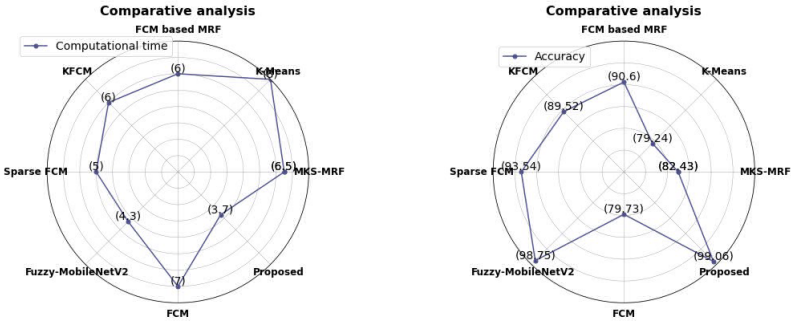


Figure 6: Differentiation of Accuracy and Computation over existing

Table 7 and Figure 6 show the differentiation between accuracy and computational time. In the present investigation, we examined the process of instance values obtained by the MapReduce framework in two situations: both with and without the suggested method. From equal-mean allocations, the time taken to execute numbers in both scenarios originates, which is the null hypothesis of this test. If the p-value is less than 0.05, which means there is a statistically significant distinction among the scenarios, then this hypothesis can be declined. With p-values of 0.0000, which indicates that the distinction is significantly different and shows that using the suggested clustering method enhances MapReduce efficiency, the null hypothesis may be dismissed based on the results of the Wilcoxon test. Figure 7 compares with proposed and without proposed.

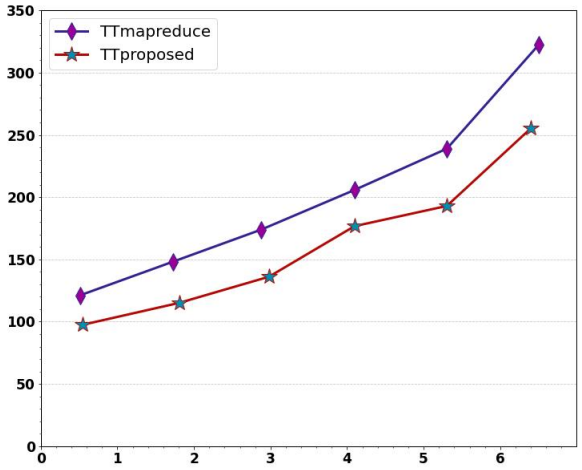


Figure 7: Effectiveness of the proposed approach

5 Conclusion and future scope

This study proposes a new clustering algorithm based on MapReduce. Utilizing traditional approaches to clustering to cluster the database gets more and more challenging as data volumes increase at an alarming rate. For an Improved Multiview Fuzzy C-Means Clustering method that is fuzzy-based, we developed a map/reduce programming that was then organized and implemented across several nodes. For our IMV-FCA technique, the general FCM is enhanced with a function parameter that estimates the distance among the instance and the CC. In addition, we present the MVBiLSTM approach, an ensemble of MobileNet V2 and three layers stacked Bidirectional LSTM to enhance performance metrics. The suggested method is more efficient for all forms of data and has the quickest execution time because of its parallelism. Furthermore, an exhaustive comparative analysis is conducted utilizing performance indicators like kappa, ARI, accuracy, NMI, recall, and precision. Future research may look into new parallelization methods like spark to reduce the computation time of the recommended approach. Additionally, a number of real-world clustering tasks involving large datasets, like Twitter analysis, satellite image analysis, and video analysis, could benefit from the implementation of the proposed approach.

Declaration

Ethics approval: This material is the authors' own original work, which has not been previously published elsewhere. The paper reflects the authors' own research and analysis in a truthful and complete manner.

Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere.

Funding: No funding was received to assist with the preparation of this manuscript.

Availability of data and material: Data will be available when requested.

Authors' contributions: The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] R. M. Alguliyev, R. M. Aliguliyev, L. V. Sukhostat, *Parallel batch k-means for big data clustering*, Computers and Industrial Engineering, **152** (2021), 107023. https://doi.org/10.1016/j.cie.2020.107023

[2] R. A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, M. A. Amanullah, I. Abaker Targio Hashem, E. Ahmed, M. Imran, *Clustering-based real-time anomaly detection-A breakthrough in big data technologies*, Transactions on Emerging Telecommunications Technologies, **33**(8) (2022), e3647. https://doi.org/10.1002/ett.3647

[3] F. H. Awad, M. M. Hamad, L. Alzubaidi, *Robust classification and detection of big medical data using advanced parallel K-means clustering, YOLOv4, and logistic regression*, Life, **13**(3) (2023), 691. https://doi.org/10.3390/life13030691

[4] O. Azeroual, A. Nikiforova, *Apache spark and mllib-based intrusion detection system or how the big data technologies can secure the data*, Information, **13**(2) (2022), 58. https://doi.org/10.3390/info13020058

[5] L. Deng, L. Wan, J. Guo, *Research on security anomaly detection for big data platforms based on quantum optimization clustering*, Mathematical Problems in Engineering, **2**(13) (2022). https://doi.org/10.1155/2022/4805035

[6] K. G. Flores, S. E. Garza, *Density peaks clustering with gap-based automatic center detection*, Knowledge-Based Systems, **206** (2020), 106350. https://doi.org/10.1016/j.knosys.2020.106350

[7] S. Garg, K. Kaur, S. Batra, G. S. Aujla, G. Morgan, N. Kumar, R. Ranjan, *En-ABC: An ensemble artificial bee colony based anomaly detection scheme for cloud environment*, Journal of Parallel and Distributed Computing, **135** (2020), 219-233. https://doi.org/10.1016/j.jpdc.2019.09.013

[8] M. Ianni, E. Masciari, G. M. Mazzeo, M. Mezzanzanica, C. Zaniolo, *Fast and effective big data exploration by clustering*, Future Generation Computer Systems, **102** (2020), 84-94. https://doi.org/10.1016/j.future.2019.07.077

[9] M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, J. Heming, *K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data*, Information Sciences, **622** (2023), 178-210. https://doi.org/10.1016/j.ins.2022.11.139

[10] F. Jemili, *Intelligent intrusion detection based on fuzzy big data classification*, Cluster Computing, **26**(6) (2023), 3719-3736. https://doi.org/10.1007/s10586-022-03769-y

[11] T. Li, G. Kou, Y. Peng, S. Y. Philip, *An integrated cluster detection, optimization, and interpretation approach for financial data*, IEEE Transactions on Cybernetics, **52**(12) (2021), 13848-13861. https://doi.org/10.1109/TCYB.2021.3109066

[12] W. Liu, *Computer network confidential information security based on big data clustering algorithm*, Wireless Communications and Mobile Computing, **2**(12) (2022). https://doi.org/10.1155/2022/4927504

[13] X. Liu, Q. Sun, W. Lu, C. Wu, H. Ding, *Big-data-based intelligent spectrum sensing for heterogeneous spectrum communications in 5G*, IEEE Wireless Communications, **27**(5) (2020), 67-73. https://doi.org/10.1109/MWC.001.1900493

[14] M. Mahdavisharif, S. Jamali, R. Fotohi, *Big data-aware intrusion detection system in communication networks: A deep learning approach*, Journal of Grid Computing, **19** (2021), 1-28. https://doi.org/10.1007/s10723-021-09581-z

[15] G. Manogaran, P. M. Shakeel, S. Baskar, C. H. Hsu, S. N. Kadry, R. Sundarasekar, B. A. Muthu, *FDM: Fuzzy-optimized data management technique for improving big data analytics*, IEEE Transactions on Fuzzy Systems, **29**(1) (2020), 177-185. https://doi.org/10.1109/TFUZZ.2020.3016346

[16] V. D. Minh, T. T. Ngan, T. M. Tuan, V. T. Duong, N. T. Cuong, *An improvement in integrating clustering method and neural network to extract rules and application in diagnosis support*, Iranian Journal of Fuzzy Systems, **19**(5) (2022), 147-165. https://doi.org/10.22111/IJFS.2022.7162

[17] S. H. Mousavi, M. Khansari, R. Rahmani, *A fully scalable big data framework for Botnet detection based on network traffic analysis*, Information Sciences, **512** (2020), 629-640. https://doi.org/10.1016/j.ins.2019.10.018

[18] A. Nowak-Brzezińska, W. Łazarz, *Qualitative data clustering to detect outliers*, Entropy, **23**(7) (2021), 869. https://doi.org/10.3390/e23070869

[19] K. K. Pandey, D. Shukla, *Stratified linear systematic sampling based clustering approach for detection of financial risk group by mining of big data*, International Journal of System Assurance Engineering and Management, (2022), 1-15. https://doi.org/10.1007/s13198-021-01424-0

[20] K. K. Pandey, D. Shukla, *Cancer tissue detection using improved K-means initialization method for multi-dimensional microarray big data clustering*, Journal of Ambient Intelligence and Humanized Computing, **14**(7) (2023), 9277-9303. https://doi.org/10.1007/s12652-022-04428-1

[21] K. Rajendra Prasad, M. Mohammed, L. V. Narasimha Prasad, D. K. Anguraj, *An efficient sampling-based visualization technique for big data clustering with crisp partitions*, Distributed and Parallel Databases, **39** (2021), 813-832. https://doi.org/10.1007/s10619-021-07324-3

[22] M. I. Razzak, M. Imran, G. Xu, *Big data analytics for preventive medicine*, Neural Computing and Applications, **32** (2020), 4417-4451. https://doi.org/10.1007/s00521-019-04095-y

[23] K. Tripathi, K. Sharma, M. Bala, *A novel clustering method using enhanced grey wolf optimizer and MapReduce*, Big Data Research, **14** (2018), 93-100. https://doi.org/10.1016/j.bdr.2018.05.002

[24] C. Xia, *Financial security risk detection in colleges and universities relying on big data clustering center scheduling algorithm*, Advances in Multimedia, **10**(1) (2022), 1-12. https://doi.org/10.1155/2022/1361041

[25] Y. Zhang, *DBSCAN clustering algorithm based on big data is applied in network information security detection*, Security and Communication Networks, **1**(10) (2022). https://doi.org/10.1155/2022/9951609

[26] W. Zhong, N. Yu, C. Ai, *Applying big data based deep learning system to intrusion detection*, Big Data Mining and Analytics, **3**(3) (2020), 181-195. https://doi.org/10.26599/BDMA.2020.9020003

[27] H. Zhou, G. Sun, S. Fu, L. Wang, J. Hu, Y. Gao, *Internet financial fraud detection based on a distributed big data approach with node2vec*, IEEE Access, **9** (2021), 43378-43386. https://doi.org/10.1109/ACCESS.2021.3062467

[28] Q. Zhu, L. Sun, *Big data-driven anomaly detection for cellular networks*, IEEE Access, **8** (2020), 31398-31408. https://doi.org/10.1109/ACCESS.2020.2973214