

Apache Spark based distributed clustering for big data analytic with application to 3D road network

Rotsnarani Sethy, Soumya Ranjan Mahanta, Mrutyunjaya Panda

Department of Computer Science and Applications, Utkal University, Bhubaneswar, India

Article Info

Article history:

Received Jun 11, 2024

Revised Sep 8, 2024

Accepted Sep 29, 2024

Keywords:

3D road network
Apache Spark
ArcGIS data visualization
Clustering
Clustering accuracy
Silhouette score
Spark-based clustering

ABSTRACT

The vast amount of data stored nowadays has turned big data analytics into a very promising research field. Clustering is an essential step in data analysis, widely used for classification, collecting statistics, and acquiring insights in specific domains of knowledge. However, the most of existing algorithms based on Lloyd-Forgy's method, have an enormously huge average-case complexity while clustering data sets with a large number of features, which may be superpolynomial time (NP-hard) and are severely constrained in terms of speed, productivity, and adaptability. Aiming to improve Lloyd-Forgy's clustering performance, K-means++ algorithms, a variety of algorithm-level optimizations which is not been well-studied, is discussed along with very promising gaussian mixture model (GMM) and soft clustering based Fuzzy C-means (FCM). Further, for fast and distributed data processing and to leverage the benefits of big data platforms, such as Apache Spark, Spark-based clustering methods are applied on three-dimensional (3D) road network data set which is collected from UCI repository. However, Spark-based clustering research is still in infancy. The distributed computation tests are conducted by allocating two core processors and one databricks unit (DBU) with 15 GB memory and measuring execution times, as well as root mean square error (RMSE), mean absolute error (MAE), clustering accuracy, and silhouette values. The results are promising and provide new research directions in the field of spark-based clustering on big data.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mrutyunjaya Panda

Department of Computer Science and Applications, Utkal University

Vani Vihar, Bhubaneswar, Odisha-751004, India

Email: mrutyunjaya74@gmail.com

1. INTRODUCTION

It is envisioned that by the end of 2050, around 70% of the world population might be living in the urban areas. This leads to developing a smart city with a lot of planning and technology as the foundation. Urban or smart city road networks, an important component of national infrastructure is now represented through intelligent geographical information system (GIS) tools like three dimensional (3D) maps for better visualization [1]. This way, 3D road network representations are helping government to have efficient planning and development of the urban/ smart city likes that of building a world-class greenfield smart city in Gurugram, Haryana in India. 3D road networks are amongst the vitally necessary elements of a smart city and intelligent transportation mechanism, which has been reconnoitering in various ways to minimize the loss of revenue by the government due to improper planning and development of the city including: damaged roads, disputed constructions, disaster management, to name a few. However, researchers still face challenges extracting 3D networks on a large scale with limited resources to monitor effectively every

construction site and/or supervise every new road [2]. It is also pointed out by the researchers that the improper planning of road transportation plays a key role in global climate change that has a negative repercussion on public health at large [3], which could have been avoided to a great extent through target fuel savings and reduced greenhouse gas emissions and other selected air pollutants. The 3D spatial map of a smart city road network with eco-routing can develop an intelligent transportation system [4] with a saving in fuel cost of 8-12% in comparison to a standard routing based 2D model [5] and can be applied for accurate prediction of fuel consumption [6]. Further, 3D road network modelling can be effectively used in internet of vehicles (IoV) scenario by providing more accurate topology for the road network to predict the vehicle's precise location with real-time position information and improved wireless channel estimation [7]. Several researchers experimented the eco-routing method on 3D spatial road network data by using the driver's driving scenario including vehicle speed, time, number of start, and stops by applying brakes. In an urban environment to study the carbon emission effects and develop models to reduce them [8]. Chen *et al.* [2] studied the multi-source 3D road network extraction method to obtain a complete and accurate road elevation data set, where they obtained the final root mean square error (RMSE) of 3.80 m and mean absolute error (MAE) of 1.94 m while comparing with those of unmanned aerial vehicle (UAV) digital surface model (DSM), light detection and ranging (LiDAR) point cloud and google earth in three cities of different scale. Next, to support the road data in 3D road network, there are four conventional way of road information extraction are available in literature such as: global positioning system (GPS) trajectory clustering methods [9], area methods [10], knowledge models [11], and artificial mapping methods [12].

Recently, deep learning techniques are used by some researchers to extract high-resolution road information from the remote sensing images using google earth and to classify the road image with road and no-road classes, which seems to be computational intensive [13]. Semantic image segmentation methods are employed by Wang *et al.* [14] on road scenes to extract exact road information while U-Net, a loss-function based optimizer network is used in [15] to extract the urban roads without object barrier. In the era of big data, new systems are to be developed to efficiently deal with the data storing and data processing [16]. Spark initially developed by the University of California Berkeley R&D Lab is an open-source tool used as an interface to exploit fault-tolerant clusters based on parallelization of the huge data for distributed computing. As defined by Databricks, which is one of the major contributors to Apache Spark, Spark based distributed computing is one of the very fast and in-memory data processing engines to process the data in real time in batch mode [17]. To conquer the defiance posed in processing the big data, use of clustering techniques are a reasonable solution. However, there are also cases where it is observed that the traditional clustering algorithms are not found suitable for large data sets [18] like the 3D road network data sets used in this research. Using Apache Spark distributed clustering utilizing parallel processing to read and process the large data sets across multiple disks and CPU/VDU in contrast to using a single PC seems to be promising area of research in big data analytics. This way, optimized computations on each cluster node are achieved through efficient utilization of available computer resources with low computational time and produce accurate solutions [19]. Even though Hadoop and Spark both are very much popular in parallel clustering approaches to deal with big data, Spark is more efficient as it not only stores the intermediate results in memory but also eliminate the exigency for numerous disk input/output (I/O) operations [20]. This way Spark is considered to be faster (at least 3 times faster on a 100 TB of data) in comparison to Hadoop MapReduce.

The traditional hard K-means method [21] always produce hard clusters, which is found to be too expensive in many machine learning applications when an object belongs to more than one cluster, and as a result, cluster boundaries necessarily overlap. On the other hand, soft clustering based on Fuzzy set representation, like Fuzzy C-means (FCM) algorithm, allows an object to belong to multiple clusters with certain membership degree lying between 0 and 1, but at the expense of high descriptiveness of the clustering results. Then the solution to the FCM clustering was to use rough set-based clustering where the result is neither too restrictive as in traditional clustering nor too descriptive in comparison to fuzzy clustering [22]. Distributed clustering on the other hand has not yet explored fully till date, creates challenges in terms of communication overhead may use few data with as minimum synchronization as possible, but produces efficient results. It is also observed that distributed clustering using Spark handling K-means is almost 100 times faster than using Hadoop based clustering [20]. This motivates us to use Apache Spark environment for our experiments in big data analytics.

Highlights of the research are:

- This research presents an Apache Spark based clustering [23], a big data platform for dealing for distributed computations
- Apache Spark can distribute the data with large features over as many computing nodes as we can afford. Here, two core processors and one DBU (Databricks unit) with 15 GB memory are employed for the research work.

- A free version of the ArcGIS software [24] is used for better understanding of 3D road network data sets obtained from UCI repository [25] through visualization, density of the area under consideration and possible outliers in the data.
- Several clustering algorithms such as simple K-means, Apache Spark based K-means++ (an efficient parallel K-means clustering), gaussian mixture model (GMM) and FCM are proposed on a 3-D road network data set for its effective implementation.
- Experimental results show the applicability of this novel Apache Spark based distributed clustering approaches in big data analytics.
- Apache Spark based FCM clustering method achieves promising results in comparison to other existing approaches in terms of i) having quality clusters, ii) lowest execution times, and iii) cause minimum error during clustering process with 100% clustering accuracy when applied in a high-dimensional and high-sparsity data sets which can be used as 'ground-truth' validation.

The rest of the paper is organized as follows. Section 2 discusses about the materials and methods used in this research followed by experiments conducted in section 3. While experimental results and discussions are presented in section 4, the paper concludes in section 5 with future scope of the research.

2. MATERIALS AND METHODS

2.1. Dataset description

In this research, 3D road network data set covering a region of 185×135 square kilometers was collected from UCI repository by adding elevation information to the previously available 2D road network data set created in North Jutland of Denmark [25]. The added elevation values are extracted for Denmark, by using an open source massive laser scan point cloud. This 3D road network data set is used as a benchmark for fuel, carbon dioxide (CO₂) and other pollutants estimation.

Road network dataset analysis in GIS using traditional routing functions are somewhat difficult to produce optimal solutions as it results combinatorial complexity. Dealing with such a complex road network for both communication and transportation network needs some efficient ways and means to allocate and provide urban amenities to the public considering shortest route and travel demand. This may be achieved through better visualization of the road network by using available free or commercial map services such as ArcGIS. At the same time, it is also very popularly used for accurate eco-routing, cycling routing and vehicle routing for implementation of IoV. It is customary to say here that a better road network not only reduces the traffic congestion but also helps in minimizing the emission of CO gas, global warming, human resentment and road accidents to name a few. For data mining applications in satellite image processing and spatial data mining, this may also be used as a “ground-truth” validation. As the 3D road network data set doesn't contain any class labels, it is more suitable to use potential clustering algorithms to obtain some missing elevation information at any points on the road network. There are four attributes with 434,874 numbers of instances; the details are as provided below.

2.1.1. Attribute information

- i) OSM_ID: open street map ID for each road segment or edge in the graph.
- ii) Longitude: web mercaptor (google format) longitude.
- iii) Latitude: web mercaptor (google format) latitude.
- iv) Altitude: height in meters.

In this, OSM_ID is the ID assigned by open street maps to the road segments. Each (longitude, latitude, and altitude) point on a road segment (with unique OSM ID) is sorted in the same order as they appear on the road. So, a 3D-polyline can be drawn by joining points of each row for each OSM_ID road segment.

Considering the above data, a descriptive analysis is conducted to understand more detailed density distribution of the data along with outliers and clusters of points using Python and ArcGIS software [24]. A graphical representation of the distribution of layers in terms of OSM ID as number of counts presented in Figure 1. Creating a density map in Python can be a valuable way to visualize the distribution of data points within a three-dimensional space. Figure 2 presents a visualization of the 3D road network data set in terms of density for each layer in the data set, which is particularly useful for insights into the concentration of data and identifying patterns or clusters, for eco-routing by the public. Here, the input is a set of points (OSM IDs) in 3D road network, and the output is a density map that illustrates where points are most densely packed with respect to corresponding longitude, latitude and altitude. Bigger bar in the Figure 1 indicates more dense points in road network at that specific category. Further, from Figure 2, one can observe that the road network is highly dense for longitude of 9.9711008, latitude of 56.9970205 and altitude (height) of 17.0527715677876, with highest density value in comparison to the other points, urge for attention for better Eco-routing while designing for smart city and intelligent transportation purposes.

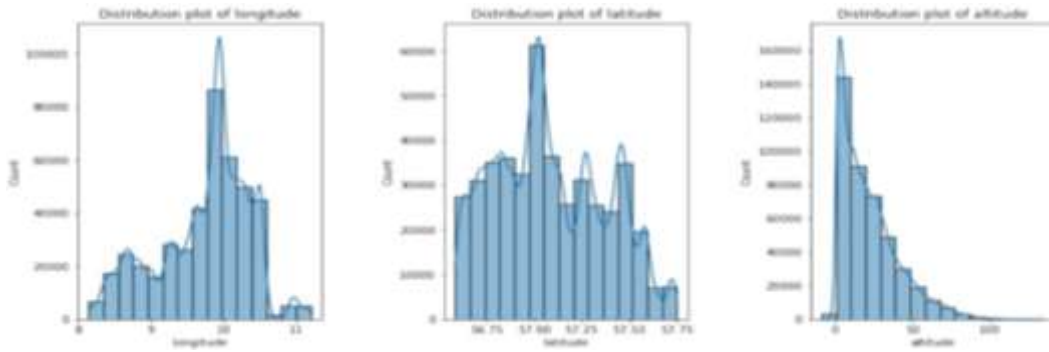


Figure 1. Distribution of 3D road network data set

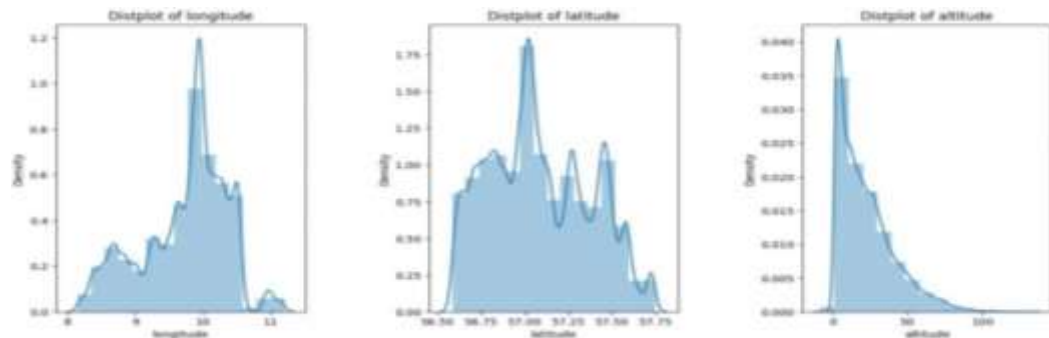


Figure 2. Density Plot for 3D road network data visualization

2.2.2. Data visualization of the road network

The 3D road network data was imported into a free version of arcGIS software which is essentially used to plot maps. 3D spatial road network data with altitude is shown in Figure 3. In Figure 3, one can observe the circles of different size represent the points with different altitude. The basemap with OSM ID and Altitude for understanding outliers are presented in Figure 4, where different circle colors represent different clusters. There are several output color categorizations used in this research to highlight the altitude of different values. In Figure 4, while pink output features are part of a cluster of high-altitude values, light blue are part of a cluster of low altitude values, red output features represent high outliers within a cluster of low altitude values and blue output features represent low outliers within a cluster of high-altitude values. Further, outlier analysis for altitude values in the 3D road network data set are performed which are shown in Figure 5 with the following description of the outlier analysis for better understanding of the data set and decision-making process.

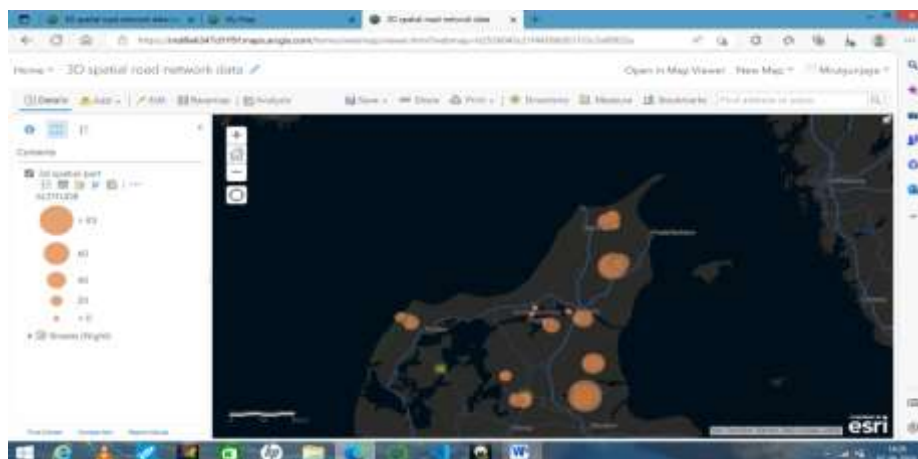


Figure 3. 3D spatial road network data with altitude

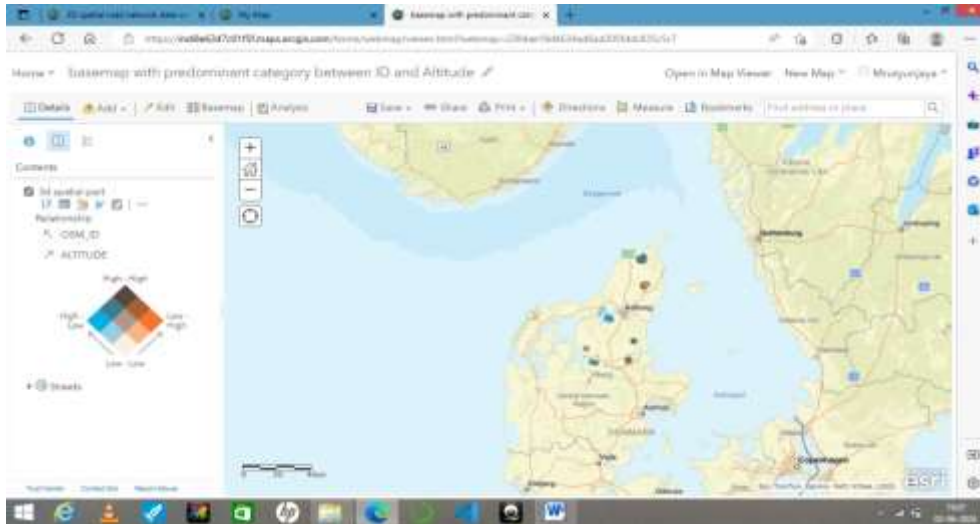


Figure 4. Basemap with OSM ID and altitude for understanding outliers

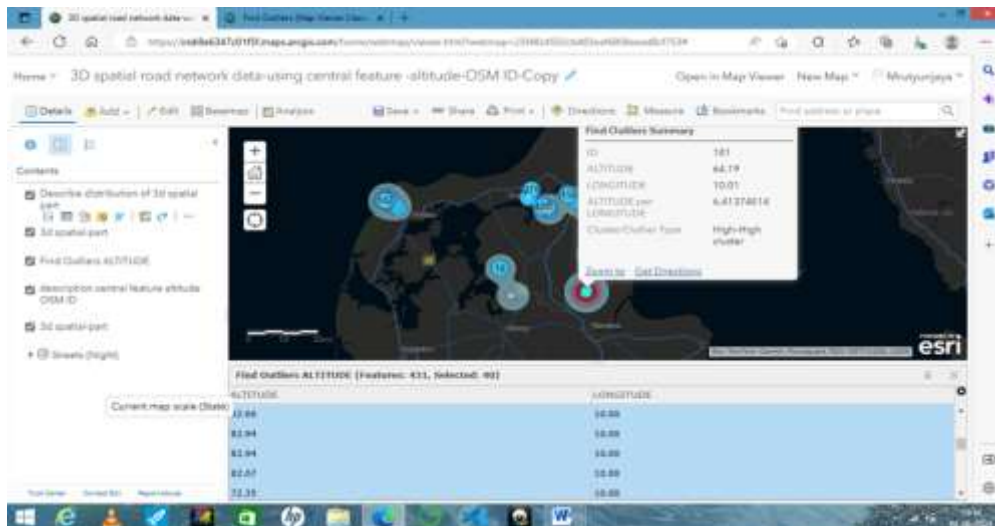


Figure 5. Outlier analysis for altitude with high-high cluster

The step wise observations recorded while performing the outlier analysis for the altitude using ArcGIS software are presented below.

- i) Step-1: initial data assessment. There were 431 valid input features. The altitude properties shown in Table 1.

Table 1. Summary statistics of altitude properties

Min	Max	Mean	Standard deviation
0.1003	8.3522	2.9549	2.1549

- There were 3 outlier locations; which will not be used to compute the optimal fixed distance band.
- ii) Step-2: scale of analysis. The optimal fixed distance band was based on the average distance to 21 nearest neighbors 2574.0000 meters.
- iii) Step-3: outlier analysis. There is 499 permutations for creating the random reference distribution for outlier analysis, out of which:
 - There are 431 output features statistically significant based on a FDR correction for multiple testing and spatial dependence.
 - There are 5 statistically significant high outlier features.

- There are 3 statistically significant low outlier features.
- There are 244 features part of statistically significant low clusters.
- There are 106 features part of statistically significant high clusters.

In the same way, outlier analysis for the other two layers (longitude and latitude) may also be conducted for better understanding of the data set. A result for outlier analysis considering longitude is shown in Figure 6, where the cluster point of interest shows not significant outlier present. In overall analysis of Figure 6, it is observed that there are 0 statistically significant high outlier features, 2 statistically significant low outlier features, 0 features part of statistically significant low clusters and 0 features part of statistically significant high clusters.

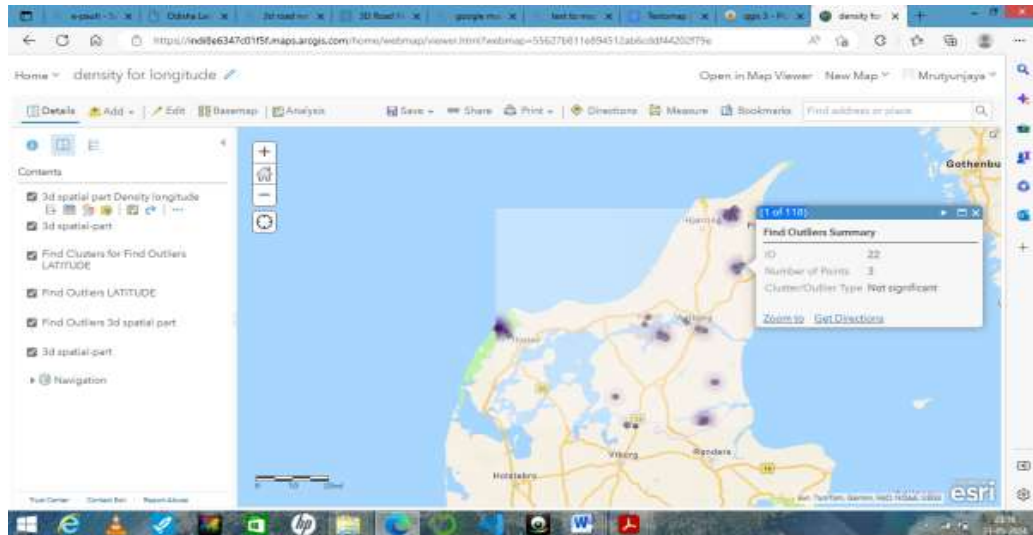


Figure 6. Outlier analysis for longitude in 3D road network data set

2.2. Proposed method

In this section, four clustering methods such as K-means, K-means++, GMM and FCM are discussed which are implemented in this research study.

2.2.1. K-means++ clustering

The K-means clustering initially proposed by Mac Queen is one of the most popular, simple, yet an efficient and stable clustering method available in the literature. Still, it has some limitations for being blocked locally based on the randomly chosen initial cluster centers. This drawback is eliminated in K-means++ by intelligently choosing a set of initial cluster centers in place of randomly chosen ones. This way, K-means++ ensures an efficient way of choosing cluster centers (centroids) with acceptable cluster quality. Distributed K-means++ is implemented in this research in ApacheSpark DataBricks environment, which is sometimes called as K-means|| (or parallel K-means) in PySpark. Parallelism is needed to address the big data analysis using machine learning algorithms so that the results can be obtained in a shorter span of time. The K-means++ [26] can find a center set that achieves $O(\log K)$ approximation for the optimal center set. At the same time, it has a computational complexity of $O(K)$, which indicates that the clustering process is slow initially for large value of K which eventually have no improvement even though less iterations are used and achieves constant bi-criteria approximation with constant probability [27]. Further, to understand the optimal cluster numbers (K) and cluster quality, elbow method with silhouette score (S) are used in this research.

Silhouette score (S) is a measure of degree of similarity between two cluster points present in the same cluster, which can be represented as in (1).

$$S_i = \frac{p_i - m_i}{\max(p_i - m_i)} \quad (1)$$

Here, p_i is the minimum average distance from i^{th} cluster point to cluster points in a different cluster and m_i represents the average distance from i^{th} cluster point to other cluster points in the same cluster. The range of

values for S_i lies between -1 to +1, where a high value of S_i indicates that i^{th} cluster point is well matched to its own cluster but poorly matched to the points in the inter clusters.

2.2.2. Gaussian mixture model

The GMM is a powerful clustering method for identifying subtle cluster patterns where it expresses the gaussian probabilistic distribution of data points, divulge hidden structures and linkages within clusters [28]. The popularity of GMM lies in its adaptability in dealing with finding clusters of different shapes, size and densities, which makes an ideal choice for the 3D road network data set used in this research. One disadvantage of GMM is in data generation from a mixture of gaussian probability distributions, which may not be possible always in real-world environments. The computational complexity of GMM depends on the number of iterations and time to compute expectations and maximization steps [29].

2.2.3. Fuzzy C-means

The fuzzy clustering method [30] applies to the points of region where it is little difficult to categorize them exactly in two clusters. FCM algorithm is a soft clustering technique using the concept of fuzzy logic so that the objects of classifications can fall in more than one cluster. Based on these separations, repetitions are performed on each cluster groups and then new centroids are calculated with the help of fuzzy partition which finally enables us to obtain the classification tasks that has linear distribution of feature space. For the data set having nonlinear distribution of feature space, kernel based FCM [31] are suggested which ultimately improves the Euclidean distance measure to have better clustering process. However, even though FCM is popular and simple to use, this method is not found suitable for clustering big data due to their complex structure [32]. Further, the drawbacks lies efficient cluster center initialization, as improper initialization can result in slow or non-optimal convergence making the cluster of low quality [32]. On the other hand, when dealing with high dimensional data, random sampling method is proposed by which data are divided into subgroups, each containing a small sample of whole data set, taking less iterations, faster convergence and with good clustering quality [33].

2.3. Performance measures

In this research, we use elbow method with silhouette score to obtain the optimal cluster and cluster quality. Further, based on the cluster size, several clustering algorithms (K-means, K-means++, GMM and FCM) were modeled on 3D road network data set to obtain the performance comparison thorough clustering time, root mean square error, mean absolute error and clustering accuracy.

3. EXPERIMENTS

All the experiments in this research are conducted in an Intel Core i5 machine with 1 TB HDD, 8 GB RAM and 2.64 GB CPU in Apache Spark environment using Databricks. Databricks presents us in terms of virtual machines (VMs) and DBUs, based on the VM instance selected. A DBU is a unit of processing capability, billed on a per-second usage in commercial version. However, we used a free databricks version with 2 cores, 1DBU and 15 GB free memory for our experimental setup in Apache Spark environment to carry distributed processing on the data set. For better understanding about this research, the steps carried out in the experimental process are presented in Figure 7 with following steps.

- Step-1: at first, we used the input 3-D road network dataset collected from UCI repository into the ArcGIS software to understand the data visualization with density distributions and then perform the outlier analysis to better understand about the road conditions that might be useful for decision making in eco-routing and smart city scenario.
- Step-2: secondly, for performing distributed data clustering approach, the input data set was used in the Apache Spark server in DataBricks environment.
- Step-3: then, schema is generated and the data is formatted using vector assembler into vectors which were used as features for analysis in 3rd step.
- Step-4: next, we used silhouette score and inertia method to find the optimal number of clusters in 4th step.
- Step-5: in the fifth step, the proposed distributed clustering model using K-means, K-means++, GMM and FCM were used for big data analytic.
- Step-6: finally, in the 6th step, experimental results are obtained using performance metrics such as: clustering time, RMSE, MAE, and clustering accuracy. and comparison with other available research is carried out to understand the suitability of the proposed research.

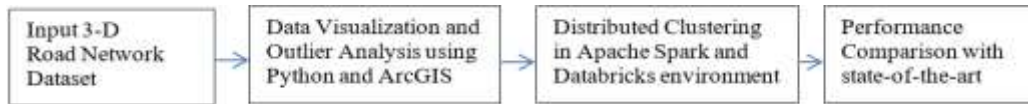


Figure 7. Experimental framework for distributed clustering for big data analytic

In this research, the number of clusters is chosen at the elbow point of the graph; similar to that of elbow point of the arm, hence the naming is done accordingly as “elbow criteria”. This elbow can’t always be unambiguously identified. The point generally present us the number of cluster point where the model can perform well. On the other hand, Inertia represents the sum of squared distances of samples to their closest cluster centers. However, we do not have always a clear clustered data. The optimal number of clusters (K) can be found through sum of squared distances within clusters (WCSS) which is calculated for different values of K. The “elbow” in the plot of WCSS against K typically indicates the optimal number of clusters to choose.

Model evaluation: silhouette score is used to assess the quality of clusters produced by various clustering techniques like K-means, K-means++, GMM and FCM. The silhouette score obtained for the whole data set is the mean value of the Silhouette Scores of all data points.

4. RESULTS AND DISCUSSION

This section presents the results obtain from the experiments conducted and discusses them to get better knowledge representation in clustering process on the 3D road network data set, for effective decision making. Figure 8 shows the distribution of data points through 3D spatial network clustering using K-means clustering for 5 clusters (K=5) in terms of three layers such as longitude, latitude, and altitude. Next, elbow method is used to find the optimal cluster, which is at K=4 as can be observed from Figure 9. In this, the silhouette score for 4 clusters is 0.427 (restricting to three decimal places). Inertia method is employed with respect to number of clusters in Figure 9 to obtain the optimal cluster value and respective cluster quality using K-means++. Even though inertia method is a very popular one in finding optimal clusters, it poses some concerns in right selection of K as it decreases monotonically with respect to number of clusters (K). However, a universal solution to find the right number of clusters in clustering process is not yet achieved so far [34].

A comparison of the clustering methods using simple K-means clustering along with Apache Spark based distributed clustering with K-means++, GMM and FCM with their respective silhouette score is given in Table 2. From Table 2, one can observe that optimal cluster is found at K=4 for all cases (0.5780 for simple K-means; 0.4298 for Apache Spark K-means++; 0.4220 for Apache Spark GMM and 0.4205 for Apache Spark FCM) having largest silhouette score than that of other values of K. At the next stage, we applied several clustering algorithms on the 3D road network data sets with different K values to understand the effectiveness of the clustering algorithm in clustering process. Root mean square error (RMSE), mean absolute error (MAE), clustering time and total execution time in seconds are used to evaluate the clustering models, which is presented in Table 3.

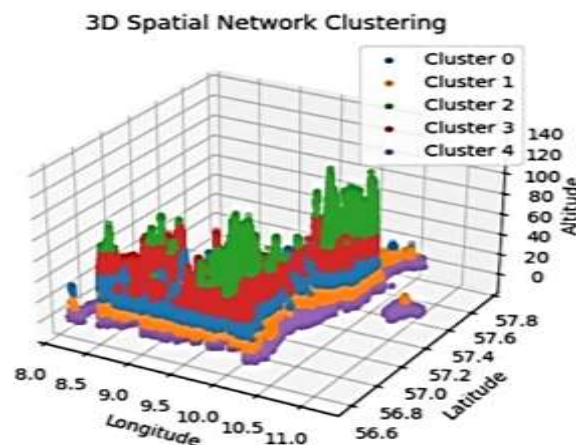


Figure 8. Distribution of data in 3D road network data set using K-means clustering (K=5)

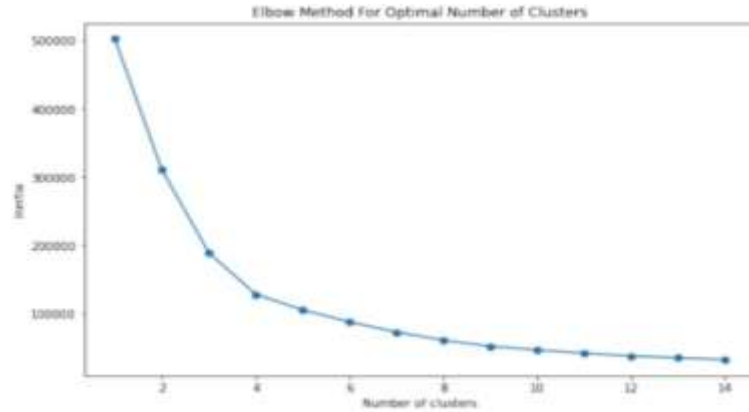


Figure 9. Plotting the inertia to find the optimal number of clusters using K-means++

It can be observed from Table 3 that after applying K-means++ in Apache Spark distributed clustering environment, the total execution time takes 90 times less than the simple K-means clustering for K=4, 43 times less for K=10 and 30.72 times less for K=40 Clusters. As in our case, K=4 is the optimal clusters, we can conclude that Apache Spark distributed clustering environment with K-means++ takes very less execution time in comparison to traditional environments with K-means clustering. To perform more research on the effectiveness of distributed clustering, GMM and FCM are used in Apache Spark environment on the same data set and the obtained results are presented in Table 4. From Table 4, it is observed that Apache Spark based FCM outperforms GMM by having highest clustering accuracy (100%), lowest RMSE (0.1980) and MAE (0.1552) at the expense of acceptable execution time (3156.30 seconds) and clustering time (10.32 seconds).

Finally, a comparison with other existing research methods are presented in Table 5 for K=4 and K=8 which demonstrates the superiority of our proposed Apache Spark based FCM over GMM, EM and K-Means clustering approaches with 100% clustering accuracy. From Table 5, it is observed that our proposed Apache Spark based GMM and FCM outperforms the K-means and EM clustering approaches in terms of % accuracy with 95.69%, 100% with optimal cluster K=4 respectively. Whereas, for K=8, FCM is most accurate with 100% accuracy in comparison to 87.39% for GMM, 90.92% for K-Means [35] and 91.67% for EM clustering [35].

From all the comparisons presented in Tables 3 to 5, it is quite evident that Apache Spark based FCM clustering is the winner followed by GMM and then K-means++, with highest clustering accuracy (100% for 4, 10, and 40 clusters) with lowest RMSE (0.1980, 0.1300, 0.0554 for 4, 10, and 40 clusters respectively) with acceptable MAE (0.1552, 0.0992, and 0.0424 for 4, 10, and 40 clusters respectively) in comparison to K-means++ but takes little more execution time than K-means++, but less than GMM in distributed clustering process.

Table 2. Silhouette score comparison for traditional and distributed clustering

Algorithm	Number of clusters	Silhouette score
Simple K-means	4	0.5780
	10	0.5085
	40	0.3437
Apache Spark K-means++	4	0.4289
	10	0.3743
	40	0.3857
Apache Spark GMM	4	0.4220
	10	0.3239
	40	0.3523
Apache Spark FCM	4	0.4205
	10	0.3698
	40	0.3917

After extensive experiments, it can be understood here that sometimes, Apache Spark takes more time for clustering process, as there are too many concurrent tasks running which is a beneficial feature as it provides fine-grained sharing and eventually leads to maximum utilization while cutting down the query latency. At the same time, difficulty in Apache Spark implementations by non-experts in machine learning

and deep learning applications with better graphical representations is a real challenge which are to be addressed in the future [36]. Still, the machine learning library of Apache Spark considered being scalable and more accessible in comparison to its counterpart Apache Hadoop MapReduce while dealing with big data of hundreds of TB or more, which is quite interesting [20]. This research is not yet fully explored; hence further and in-depth studies may be needed to confirm its suitability especially in developing eco-routing model while designing an IoV, smart city and intelligent transportation scenario for a sustainable environment.

Table 3. Experimental results using clustering algorithm in Apache Spark environment

Algorithm/error measures	No of clusters	RMSE	MAE	Clustering time in Sec	Total execution time in sec
Simple K-means	4	3.2201	1.7336	4.7395	2089.5237
	10	1.4028	0.8711	5.6595	2090.5515
	40	0.5157	0.3973	13.6936	2078.3555
Apache Spark K-means++	4	2.6011	1.4158	4.4808	22.9585
	10	1.4028	0.8711	9.9141	47.2690
	40	0.5157	0.3973	30.9308	65.5233

Table 4. Comparison of GMM and FCM in Apache Spark on 3D road network data set

Algorithm/error measures	No of clusters	RMSE	MAE	Clustering time in sec	Total execution time in sec	% Accuracy
Apache Spark GMM	4	0.1997	0.1555	2.64	3291.46	95.69
	10	0.1364	0.1046	30.67	3256.24	86.71
	40	0.0598	0.0460	74.57	3362.70	86.54
Apache Spark FCM	4	0.1980	0.1552	10.32	3156.30	100
	10	0.1300	0.0992	97.29	3268.54	100
	40	0.0554	0.0424	843.67	3453.16	100

Table 5. Comparison with others work applied on 3D road network data set

Algorithm	K (no. of clusters)	Achieved clustering accuracy %
K-means [35]	4	91.67
K-means [35]	8	90.92
EM [35]	4	90.71
EM [35]	8	91.67
Ours (GMM)	4	95.69
Ours (GMM)	8	87.39
Ours (FCM)	4	100
Ours (FCM)	8	100

5. CONCLUSION

In this research, the aim was to develop an efficient distributed clustering model for big data analytics using 3D road network data set. For any data analytic process, data visualization plays a very vital role. So, data distribution with density plot and pair plot is used to understand the relations amongst the variables. Further, to check whether any outlier is present in the data set, experiments are conducted using ArcGIS to achieve this. Next, the data set is used on an Apache Spark based environment using distributed clustering through K-means++ (parallel K-means), GMM and FCM and the performance of each one of them are recorded in terms of Silhouette score, clustering time, total execution time, RMSE, MAE, and clustering accuracy.

From the comparison, it is observed that the distributed clustering through Apache Spark based K-means++ outperform traditional K-means clustering by taking 90 times less time with more accurate clustering. Next, to perform further investigation about the effectiveness of the clustering process, several other clustering process such as GMM and FCM are employed in Apache Spark distributed environment. After comparing the results obtained by our experiments with different clustering models and subsequently comparing with the other available research applied on the same data set, we conclude that FCM based distributed clustering is the best clustering model with highest clustering accuracy (100%), lowest RMSE (0.1980) and MAE (0.1552) at the expense of acceptable execution time (3156.30 seconds).

From the above findings, it is envisaged that this research will be helpful in warrant the future advancement in data analytics and machine learning especially clustering techniques using distributed computing approach in order to uncover the hidden information from the vast amount of most challenging unstructured and spatio-temporal data. Further, big data machine learning using the effectiveness of several

proposed and discussed clustering algorithms which are efficient in 3D road network design may pave the way for the scientists/engineers to design future smart city and intelligent transportation system. While this research will pave the way for the interested researchers and practitioners to carry out further experiments to build a suitable model of interest with possible threats to validity of the clustering process. In future, some more distributed clustering techniques in a bigger and complex data set are to be investigated to address the issues pertaining to big data analytics.




REFERENCES

- [1] F. J. Ariza-López, A. T. Mozas-Calvache, M. A. Ureña-Cámara, and P. Gil de la Vega, "Dataset of three-dimensional traces of roads," *Scientific Data*, vol. 6, no. 1, p. 142, Aug. 2019, doi: 10.1038/s41597-019-0147-x.
- [2] Y. Chen, X. Yang, L. Yang, and J. Feng, "An automatic approach to extracting large-scale three-dimensional road networks using open-source data," *Remote Sensing*, vol. 14, no. 22, p. 5746, Nov. 2022, doi: 10.3390/rs14225746.
- [3] S. W. Busho and D. Alemayehu, "Applying 3D-eco routing model to reduce environmental footprint of road transports in Addis Ababa City," *Environmental Systems Research*, vol. 9, no. 1, p. 17, Dec. 2020, doi: 10.1186/s40068-020-00179-0.
- [4] M. Kaul, B. Yang, and C. S. Jensen, "Building accurate 3D spatial networks to enable next generation intelligent transportation systems," in *2013 IEEE 14th International Conference on Mobile Data Management*, Jun. 2013, pp. 137–146, doi: 10.1109/MDM.2013.24.
- [5] G. Tavares, Z. Zsigraiova, V. Semiao, and M. G. Carvalho, "Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling," *Waste Management*, vol. 29, no. 3, pp. 1176–1185, Mar. 2009, doi: 10.1016/j.wasman.2008.07.013.
- [6] C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul, "EcoMark: evaluating models of vehicular environmental impact," in *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, Nov. 2012, pp. 269–278, doi: 10.1145/2424321.2424356.
- [7] D. Cao, J. Ru, J. Qin, A. Tolba, J. Wang, and M. Zhu, "3D road network modeling and road structure recognition in internet of vehicles," *Computer Modeling in Engineering & Sciences*, vol. 138, no. 2, pp. 1365–1384, 2024, doi: 10.32604/cmescs.2023.030260.
- [8] J. F. Coloma, M. García, Y. Wang, and A. Monzón, "Green eco-driving effects in non-congested cities," *Sustainability (Switzerland)*, vol. 10, no. 1, p. 28, Dec. 2018, doi: 10.3390/su10010028.
- [9] S. Dutta, A. Das, and B. K. Patra, "CLUSTMOSA: clustering for GPS trajectory data based on multi-objective simulated annealing to develop mobility application," *Applied Soft Computing*, vol. 130, p. 109655, Nov. 2022, doi: 10.1016/j.asoc.2022.109655.
- [10] B. Cheng, H. Ji, and Y. Wang, "A new method for constructing roads map in forest area using UAV images," *Journal of Computational Methods in Sciences and Engineering*, vol. 23, no. 2, pp. 573–587, Apr. 2023, doi: 10.3233/JCM-226621.
- [11] C. Zhang, E. Baltasvias, and A. Gruen, "Knowledge-based image analysis for 3D road reconstruction," *Asian Journal of Geoinformatics*, vol. 1, no. 4, pp. 3–14, 2001, doi: 10.3929/ethz-a-004334193.
- [12] Q. Gu, B. Xue, J. Song, X. Li, and Q. Wang, "A high-precision road network construction method based on deep learning for unmanned vehicle in open pit," *Mining, Metallurgy & Exploration*, vol. 39, no. 2, pp. 397–411, Apr. 2022, doi: 10.1007/s42461-022-00548-6.
- [13] P. Li *et al.*, "Road network extraction via deep learning and line integral convolution," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2016, pp. 1599–1602, doi: 10.1109/IGARSS.2016.7729408.
- [14] Q. Wang, J. Gao, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 230–241, Jan. 2018, doi: 10.1109/TITS.2017.2749964.
- [15] H. He, S. Wang, S. Wang, D. Yang, and X. Liu, "A road extraction method for remote sensing image based on encoder-decoder network," *Journal of Geodesy and Geoinformation Science*, vol. 3, no. 2, pp. 16–25, 2020, doi: 10.11947/j.JGGS.2020.0202.
- [16] R. Talavera-Llames, R. Pérez-Chacón, A. Troncoso, and F. Martínez-Álvarez, "Big data time series forecasting based on nearest neighbours distributed computing with Spark," *Knowledge-Based Systems*, vol. 161, pp. 12–25, Dec. 2018, doi: 10.1016/j.knsys.2018.07.026.
- [17] M. Zaharia *et al.*, "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016, doi: 10.1145/2934664.
- [18] M. D. C. Indirman, G. W. Wirriasto, and L. A. S. I. Akbar, "Distributed machine learning using HDFS and Apache Spark for big data challenges," *E3S Web of Conferences*, vol. 465, p. 02058, Dec. 2023, doi: 10.1051/e3sconf/202346502058.
- [19] S. Quddus, "Fast algorithms for unsupervised learning in large data sets," in *Computer Science & Information Technology (CS & IT)*, Jan. 2017, pp. 69–76, doi: 10.5121/csit.2017.70207.
- [20] M. Hai, Y. Zhang, and H. Li, "A Performance comparison of big data processing platform based on parallel clustering algorithms," *Procedia Computer Science*, vol. 139, pp. 127–135, 2018, doi: 10.1016/j.procs.2018.10.228.
- [21] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sep. 1999, doi: 10.1145/331499.331504.
- [22] R. Xu and D. WunschII, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005, doi: 10.1109/TNN.2005.845141.
- [23] "Apache Spark™ tutorial: getting started with Apache Spark on Databricks," [www.databricks.com](https://www.databricks.com/spark/getting-started-with-apache-spark). <https://www.databricks.com/spark/getting-started-with-apache-spark>.
- [24] "ArcGIS Online: desktop GIS Software Suite," [www.arcgis.com](https://www.arcgis.com/home/index.html). <https://www.arcgis.com/home/index.html> (accessed Jun. 09, 2024).
- [25] M. Kaul, "3D road network (North Jutland, Denmark)," *UCI Machine Learning Repository*, 2013. <https://doi.org/10.24432/C5GP51> (accessed Jun. 09, 2024).
- [26] D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, vol. 07-09-Janu, pp. 1027–1035.
- [27] A. Aggarwal, A. Deshpande, and R. Kannan, "Adaptive sampling for k-means clustering," in *International Workshop on Approximation Algorithms for Combinatorial Optimization*, 2009, pp. 15–28, doi: 10.1007/978-3-642-03685-9_2.





- [28] Y. Zhang *et al.*, “Gaussian mixture model clustering with incomplete data,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 1s, pp. 1–14, Jan. 2021, doi: 10.1145/3408318.
- [29] M. Sugiyama, “Maximum likelihood estimation for gaussian mixture model,” in *Introduction to Statistical Machine Learning*, Elsevier, 2016, pp. 157–168.
- [30] A. Guha and N. Veeranjanyulu, “Prediction of bankruptcy using big data analytic based on fuzzy C-means algorithm,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 2, pp. 168–174, Jun. 2019, doi: 10.11591/ijai.v8.i2.pp168-174.
- [31] J. Jędrzejowicz, P. Jędrzejowicz, and I. Wierzbowska, “Apache Spark Implementation of the distance-based kernel-based fuzzy c-means clustering classifier,” in *Intelligent Decision Technologies 2016: Proceedings of the 8th KES International Conference on Intelligent Decision Technologies (KES-IDT 2016)--Part I*, 2016, pp. 317–324, doi: 10.1007/978-3-319-39630-9_26.
- [32] S. E. Hashemi, F. Gholian-Jouybari, and M. Hajiaghaei-Keshteli, “A fuzzy C-means algorithm for optimizing data clustering,” *Expert Systems with Applications*, vol. 227, p. 120377, Oct. 2023, doi: 10.1016/j.eswa.2023.120377.
- [33] A. G. Di Nuovo and V. Catania, “An evolutionary fuzzy c-means approach for clustering of bio-informatics databases,” in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 2077–2082, doi: 10.1109/FUZZY.2008.4630656.
- [34] A. Rykov, R. C. De Amorim, V. Makarenkov, and B. Mirkin, “Inertia-based indices to determine the number of clusters in K-means: an experimental evaluation,” *IEEE Access*, vol. 12, pp. 11761–11773, 2024, doi: 10.1109/ACCESS.2024.3350791.
- [35] D. Li, S. Wang, N. Gao, Q. He, and Y. Yang, “Cutting the unnecessary long tail: cost-effective big data clustering in the cloud,” *IEEE Transactions on Cloud Computing*, vol. 10, no. 1, pp. 292–303, 2022, doi: 10.1109/TCC.2019.2947678.
- [36] A. Hussein Ali, M. Nawaf Abbod, M. Khamees Khaleel, M. Abdulghafoor Mohammed, and T. Sutikno, “Large scale data analysis using MLlib,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 5, pp. 1735–1746, Oct. 2021, doi: 10.12928/telkomnika.v19i5.21059.

BIOGRAPHIES OF AUTHORS







Ms. Rotsnarani Sethy     received the B.Sc. degree in Mathematics from S. B. Womens College, Cuttack, the MCA degree from North Odisha University and M.Tech (Computer Science) from Utkal University, India. She received Rajiv Gandhi national Fellowship for carrying out her Ph.D. in the Department of Computer Science and Applications, Utkal University. Her research interests include: big data analytic, machine learning, and database design. She has published 4 articles in conference and peer reviewed journals. She can be contacted at email: roshnaranimca@gmail.com.



Mr. Soumya Ranjan Mahanta     received the B.Sc. in ITM and M.Sc. in ITM from Ravenshaw University and M. Tech. (CSE) from Department of Computer Science and Applications, Utkal University, Odisha, India. Presently, he is working as Lecturer in Computer Science at DRIEMS University, Cuttack, India. His research interest include: machine learning, artificial intelligence, and Python programming. He can be contacted at email: dipusoumyaranjan019@gmail.com.



Dr. Mrutyunjaya Panda     is Associate Professor at Computer Science and Applications, Utkal University, Vani Vihar, Bhubaneswar, Odisha, India. He holds a Ph.D. degree in Computer Science, M.Engg. in Communication system Engineering and B.Engg. in Electronics and Tele-Communication Engineering and MBA in Human Resource Management. His research areas are data mining, bio-medical image processing, big data analytic, natural language processing, and social network analysis. He has authored and co-authored more than 150 research articles in reputed journals, conferences and book chapters. He has also edited 7 books and authors two text books to his credit. He is a member of editorial board and an active reviewer of many journals and conferences. He can be contacted at email: mrutyunjaya74@gmail.com.