KLE Society's
KLE Technological University, Hubballi.

A Minor Project -2 Report

On

Data Clustering Using Firefly Algorithm in Distributed Environment

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

In

School of Computer Science and Engineering

Submitted By

Trupti Venkatesh	01FE21BCS176
Aditi Khyadad	01FE21BCS072
Naveenkumar B	01FE21BCS051
Tejas N	01FE21BCS053

Under the guidance of
Prof. Shivalingappa Battur

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBBALLI – 580 031

Academic year 2023-24

KLE Society's
KLE Technological University, Hubballi.

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Minor Project -2 entitled "Data Clustering Using Firefly Algorithm in Distributed Environment" is a bonafied work carried out by the student team Trupti Venkatesh USN:01FE21BCS176 , Aditi Khyadad USN:01FE21BCS072 , Naveenkumar B USN:01FE21BCS051 , Tejas N USN:01FE21BCS053 , in partial fulfillment of completion of Sixth semester B. E. in School of Computer Science and Engineering during the year 20232024. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said program.

Guide	Head, SoCSE
Prof. Shivalinagappa Battur	Dr. Vijayalakshmi.M

External Viva -Voce:

Name of the Examiners	Signature with date
1.	
2.	

Acknowledgement

We would like to thank our faculty and management for their professional guidance towards the completion of the project work. We take this opportunity to thank Dr. Ashok Shettar, Vice-Chancellor, Dr. B.S.Anami, Registrar, and Dr. P.G Tewari, Dean Academics, KLE Technological University, Hubballi, for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Dean of Faculty, SoCSE and Dr. Vijayalakshmi M, Professor and Head, SoCSE for having provided us direction and facilitated for enhancement of skills and academic growth.

We thank our guide Prof. Shivalingappa Battur, Assisstant Professor, SoCSE for the constant guidance during interaction and reviews.

We extend our acknowledgement to the reviewers for critical suggestions and inputs. We also thank Project Co-ordinator Dr. Uday Kulkarni, and reviewers for their suggestions during the course of completion.

We express gratitude to our beloved parents for constant encouragement and support.

Naveenkumar Baraker - 01FE21BCS051
Tejas C. Nadagadalli - 01FE21BCS053
Aditi Khyadad - 01FE21BCS072
Trupti Venkatesh - 01FE21BCS176

ABSTRACT

In the era of digital transformation, big data is a valuable asset, enabling businesses to innovate and make effective decisions. However, the volume, velocity, and variety of big data present significant challenges in storage, processing, and analysis. This research enhances traditional clustering techniques by integrating the Firefly Algorithm with K-Means to optimize the number of clusters, thereby improving accuracy and scalability in big data environments. The hybrid approach was evaluated on the PokerHand and Tabular Playground datasets using a multi-node Apache Hadoop cluster on AWS EC2. The evaluation metrics included clustering purity, execution time, and computational efficiency. The hybrid approach demonstrates superior clustering purity and robustness compared to traditional methods, despite some variability, and shows marked improvements in execution time and resource utilization. The results underscore the potential of combining clustering algorithms with optimization techniques for efficient large-scale data categorization, paving the way for real-time data processing applications. Future work aims at further refinements, exploring additional hybrid methods, and extending the approach to other big data frameworks

Keywords : *Firefly Algorithm, K-Means Clustering, Elbow Method, Distributed Computing, WCSS, Clustering, PokerHand.*

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In the present time of digital alteration, many trades have found out that big data is a very profitable asset because it allows them to be more innovative and effective in whatever they do. When individuals refer to 'Big Data', they are talking about a massive amount of various kinds of structured and unstructured information being produced at incredibly high speeds from sources like social media posts, sensor readings or transaction records. Being able to use all this can give you an upper hand by enabling quick decision making based on valuable insights that may otherwise go unnoticed. Nonetheless there are huge difficulties associated with storing processing and analyzing these massive amounts of information due mainly on account of its size complexity and rate at which it is being produced.

1.1 Motivation

Dealing with unlabelled data is one of the biggest challenges of big data analytics. Unlabelled data does not have any pre-assigned categories or classes. Clustering methods come into play at this point. Clustering is an unsupervised learning task that aims at dividing a dataset into groups so that each object within the same group is more similar to one another than those in other groups. This helps in understanding and analyzing the data better by making it easier for analysts to see patterns among like items. However, conventional clustering algorithms do not scale well when dealing with mixed types of data or large amounts of information since they are limited in terms of scalability and adaptability. To combat these shortcomings, we need a method that can handle mixed data types and scale to large datasets. It can be accomplished through clustering.

1.2 Literature Review / Survey

Traditional clustering algorithms such as K-Means, hierarchical clustering, and DBSCAN are effective but struggle with big data due to issues like sensitivity to initial conditions and computational intensity. Optimization algorithms like the Firefly Algorithm and Genetic Algorithms improve clustering by dynamically optimizing results. Hybrid approaches combine these with traditional methods for better accuracy and scalability.

1.2.1 Traditional Clustering Methods

Traditional clustering algorithms, such as K-Means, hierarchical clustering, and DBSCAN, are widely used due to their simplicity and effectiveness in many applications. However, these methods often struggle with the computational demands of big data [1]. The k-means algorithm, for example, is efficient but can be sensitive to the initial selection of centroids and is not well-suited for non-globular clusters. Hierarchical clustering provides a detailed hierarchy of clusters but is computationally intensive, making it impractical for very large datasets. DBSCAN can identify clusters of arbitrary shape but requires careful tuning of its parameters and is less effective in high-dimensional spaces. DBSCAN can identify clusters of arbitrary shape but requires careful tuning of its parameters and is less effective in high-dimensional spaces.

1.2.2 Optimization Algorithms

In order to achieve a meaningful improvement, the use of traditional methods is not sufficient. In solving complex optimization problems, algorithms such as the Firefly Algorithm, Genetic Algorithms, and Particle Swarm Optimization [2] have been adopted as some of the optimization algorithms. Nature-inspired algorithms imitate natural processes. This algorithm can dynamically optimize clustering results by changing the brightness and attractiveness of the fireflies so as to deal with large and complex data sets.

1.2.3 Hybrid Approaches

Traditional clustering and optimization techniques have attracted a lot of attention lately. By combining these two methods, hybrid approaches try to improve the accuracy and scalability of clusters. For instance, one might apply the K-Means algorithm to the Firefly Algorithm for an initial clustering through k-means, followed by optimizing cluster centroids using the Firefly Algorithm [3]. Consequently, what happens with such a combined approach is that it ensures stability in clustering and produces better final clusters.

1.2.4 Consensus Clustering Methods

Clustering or clustering ensemble, conventional methods include combining various clustering outcomes in order to achieve stable and consistent clustering. These methods can be divided into two strategies: ECIO (Implicit Clusterings Intend Optimization) and ECEO (Explicit Clusterings Expectation Optimization). ECIO methods do not establish any global objective function, but instead use heuristics like co-association matrices and genetic algorithms for approximating solutions. In contrast to this, ECEO methods employ global objective functions

such as those found in expectation maximization or non-negative matrix factorization algorithms which guide clustering based on overall data structure rather than local neighborhood information only [4]. However, they may have difficulties in balancing between interpretability and efficiency when applied at large scale problems.

1.3 Problem Statement

To develop a data clustering approach using the Firefly Algorithm to improve efficiency in processing and analyzing large-scale datasets.

1.4 Objectives and Scope of the project

The primary aim of this project is to enhance the efficiency and accuracy of K-Means clustering through the implementation of the Firefly Algorithm using parallel computing techniques. The following key objectives outline the scope of this endeavor.

1.4.1 Objectives

- To determine the subprocess involved in parallel execution of the firefly algorithm.
- To design and execute the firefly algorithm using parallel computing technique.
- To compare the results with the other state-of-the-art clustering methods for the data.

1.4.2 Scope of the project

The scope of this project focuses on enhancing clustering methods to handle large and complex datasets effectively. The proposed approach integrates traditional clustering algorithms with optimization techniques and consensus clustering methods. Specifically, the method will:

1. Utilize K-Means for initial clustering, ensuring efficiency in partitioning data.
2. Apply optimization algorithms like the Firefly Algorithm to refine cluster centroids, improving accuracy and adapting to complex data structures.

Boundaries of the Method

1. Effective for large datasets but may require substantial computational resources for extremely large-scale data.
2. While optimized for high-dimensional data, performance may degrade with extremely high dimensions due to the curse of dimensionality.

3. Dependent on available computational power; high-performance computing environments are preferable for best results.

Chapter 2

REQUIREMENT ANALYSIS

Requirement analysis is a crucial phase in the development of the Distributed Firefly Optimization and K-means Clustering system. It involves identifying and documenting the functional and non-functional requirements necessary to meet the needs of the end-users and stakeholders. Functional requirements define the specific behavior and operations of the system, such as data handling, optimization using the Firefly Algorithm, result aggregation, determining optimal clusters, and performing K-means clustering. Non-functional requirements, on the other hand, encompass the system's performance, reliability, usability, maintainability, security, compatibility, and scalability, among others.

2.1 Functional Requirements

Functional requirements define the specific behavior and operations of the system, such as data handling, optimization using the Firefly Algorithm, result aggregation, determining optimal clusters, and performing K-means clustering.

- Data Handling and Distribution:
  - The system shall be able to handle large datasets.
  - Split the dataset D into n chunks
  - Distribute each chunk to master nodes in a distributed environment.
- Firefly Algorithm Optimization:
  - The system shall be able to apply Firefly algorithm to each data chunk in master node.
- Aggregation and Combining Results:
  - The system shall be able to aggregate the results from the master node.
- Apply Kmeans:
  - The system shall be able to apply K-means clustering using the determined optimal K and repeat the assignment of points to the closest centroid and revise centroids until convergence.

- Output Final Clusters:
  - The system shall provide the final clusters after the convergence of the K-means algorithm.

2.2 Non Functional Requirements

Non-functional requirements, on the other hand, encompass the system's performance, reliability, usability, maintainability, security, compatibility, and scalability, among others.

- The system should efficiently handle increasing sizes of datasets by utilizing distributed processing.
- The system should efficiently use computational resources to minimize costs and maximize throughput.
- Optimize algorithms to minimize energy consumption, particularly in large-scale distributed environments.

2.3 Software Requirements

The setup included making a multi-node Apache Hadoop cluster on AWS EC2 utilizing t2.medium occurrences with Amazon Linux 2 AMI. Designed a VPC for secure organizing and connected 50 GB EBS volumes to each occasion. Hadoop 3.3.1 and OpenJDK 8 were introduced, and fundamental environment factors were set. The cluster comprised one ace hub running NameNode and ResourceManager administrations and numerous laborer hubs running DataNode and NodeManager administrations. We customized setup records (core-site.xml, hdfs-site.xml, yarn-site.xml, mapred-site.xml), set up passwordless SSH, designed the HDFS, and begun YARN administrations. The cluster's health and usefulness were confirmed utilizing Hadoop's web interfacing and command-line tools, ensuring a strong environment for the tests.

Chapter 3

SYSTEM DESIGN

The project leverages the Firefly Algorithm (FA) to optimize the selection of the K-value in clustering, simulating fireflies' flashing behavior to explore and converge on the optimal number of clusters. The Elbow Method validates this optimal K-value by identifying where the explained variance diminishes. This K-value is then used in a distributed K-means clustering approach, where slave nodes independently cluster data chunks and aggregate the results into a final solution. To demonstrate scalability and performance, the method is applied to 4-node and 6-node cluster setups, showcasing its flexibility and efficiency in handling varying computational resources and illustrating the robustness of the distributed clustering approach.

3.1 Architecture Design

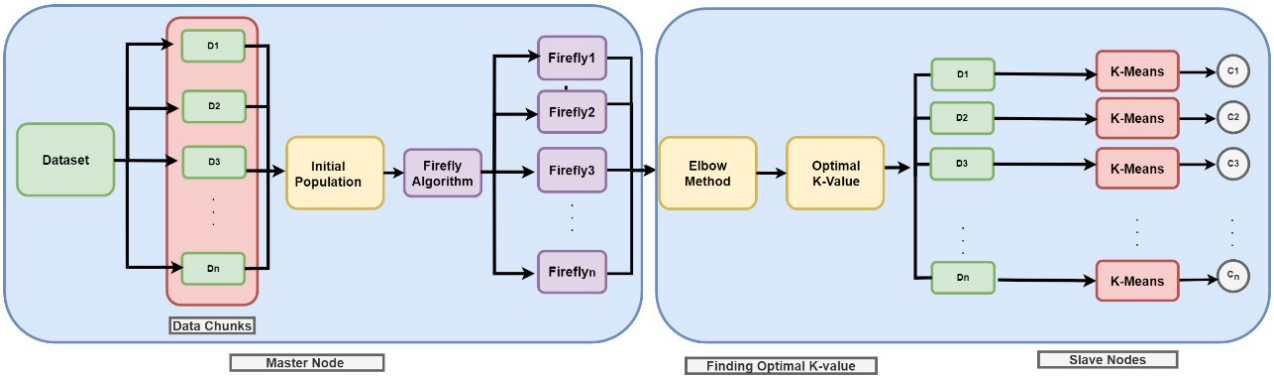

Figure 3.1: Workflow of the Firefly Algorithm and K-means Clustering in a Distributed Env

3.1.1 Firefly Algorithm

It was introduced by Xin-She Yang in 2008 [5], which is an evolutionary algorithm inspired by nature that emulates the social behavior and bioluminescent communication of fireflies. It excels in resolving optimization issues, especially those with intricate and multi-modal landscapes. The fundamental idea of the FA is the attraction between fireflies, which is determined by their brightness and is proportional to the value of the objective function. Vibrant fireflies inherently draw the attention of other brighter ones, and the less vibrant

ones gravitate toward them. This creates a dynamic equilibrium between exploitation and exploration in the search space.

Attractiveness: A firefly's β attraction is based on its brightness, which is connected to the objective function. The attractiveness is expressed as follows as it diminishes with distance r, given as:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{3.1}$$

where γ is the light absorption coefficient and β_0 is the attractiveness at $r = 0$. [5].

Movement: The following describes how a firefly i moves in the direction of a brighter firefly j:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(\text{rand} - 0.5) \tag{3.2}$$

where rand is an arbitrary number uniformly distributed in $[0, 1]$, α is the randomization dimension, and x_i and x_j are the spots of firefly i and j [5].

Update and Iteration: Firefly positions are updated iteratively using the orientation equation until a stopping condition is satisfied, like convergence or a certain number of iterations.

3.1.2 K-Means Algorithm

It is a popular approach which works by reducing the within-cluster sum of squares (WCSS) to divide a dataset into K groups. K centroids are first chosen at random among the data points. Using the Euclidean distance metric, every point of data is assigned to the closest centroid in the assignment step [6]. In this step, the cluster to which every point of data belongs is determined. The centroids are then computed as the average of all data points given to each cluster during the update stage. The centroids are assigned and updated iteratively until they settle, which means that following iterations do not substantially change their positions.

Initialization: Choose K starting centroids at random from the dataset.

Assignment: Apply the Euclidean distance equation to each data point and assign it to the closest centroid. [7]:

$$\text{Cluster, } i = \left\{x_j \mid \|x_j - \mu_i\|_2^2 \leq \|x_j - \mu_k\|_2^2 \, \forall k, 1 \leq k \leq K\right\} \tag{3.3}$$

where the centroid of cluster i is μ_i .

Update: Determine the centroids again using the average of all the data points allocated to each cluster [7]:

$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \tag{3.4}$$

where the group of points in cluster i is denoted by C_i.

Iteration: Unless the centroids no longer significantly vary, recur the assignment and update

processes.

The K-Means algorithm aims to reduce the following WCSS [1] [8]:

$$\text{WCSS} = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2 \tag{3.5}$$

3.1.3 Elbow Method

This method is used to figure out how many clusters in a dataset are best for K-Means clustering. To calculate the WCSS for every configuration, K-Means runs for a range of cluster counts, usually from 1 to 10 or more. The total squared distances between every data point and its designated centroid inside a cluster are represented by the WCSS [9]. A clear pattern frequently appears when the WCSS values are plotted against the total number of clusters, K. The "elbow" area of the plot, where the rate of WCSS is lower, dramatically slows down and adding more clusters offers diminishing rewards in terms of explaining the variance in the data, is where the optimal K is found. In order to balance model complexity and clustering efficacy, this method offers a useful guideline for determining the number of clusters that best depicts the underlying structure of the dataset.

3.2 Dataset Design

The considered datasets are as follows :

1. PokerHand - This dataset comprises of 1,025,010 instances representing poker hands. Each instance includes 10 attributes: 5 for card ranks (2-14) and 5 for suits (hearts, clubs, diamonds, spades). The target variable is the poker hand category, with 10 classes ranging the two dominant classes account for over 90% of the samples, i.e., nothing in hand with 49.9% and one pair with 42.4%.

2. Tabular PlayGround - It is a collection of structered data and consists of approximately 200,000 instances with multiple anonymized features and also contains multiple features (columns) which contains both numeric value and categorical labels.

Chapter 4

IMPLEMENTATION

4.1 Data Preprocessing and Data Partitioning

Initially, the original data set is decomposed into smaller number of segments to support the dispatch of parallel tasks. Each segment, denoted as D1,D2,. . . ,Dn, is distributed within the cluster. The master node is responsible for partitioning the dataset and distributing the data chunks to the respective slave nodes for further processing, which can be observed in the Figure An initial population of candidate solutions is generated by the master node for the Firefly Algorithm. Each candidate solution represents a potential number of clusters (K-value).

4.2 Clustering in the Distributed Environment

The FA is applied to optimize the selection of the K-value which is shown in the Algorithm 1. This bio-inspired algorithm simulates the flashing behavior of fireflies to find the optimal solution through interactions based on their brightness (objective function measure). The fireflies' interactions lead to exploration and convergence towards the optimal K-value by attracting other fireflies based on their brightness, which correlates with the quality of the clustering solution.

The Elbow Method is used to validate and determine the optimal K-value from the solutions generated by the Firefly Algorithm. This method identifies the point where the explained variance starts to diminish, indicating the most appropriate number of clusters. The K-value corresponding to the elbow point is identified as the optimal number of clusters that is need to have for the dataset.

The optimal K-value is communicated to the slave nodes, each of which is assigned a data chunk. Each slave node performs K-means clustering independently on its respective data chunk (D1, D2, ..., Dn), producing clusters (C1, C2, ..., Cn). The resulting clusters from all slave nodes are aggregated to form the final clustering solution for the entire dataset.

To validate the scalability and performance of the implementation, applied this distributed clustering approach to different cluster configurations, specifically a 4-node cluster. These configurations were chosen to demonstrate the flexibility and efficiency of the method in

handling various levels of computational resources. The results from these experiments further illustrate the robustness of our approach in distributed environments.

4.3 Algorithm

Algorithm 1 Distributed Firefly Optimization and K-means Clustering

Require: Large Dataset D
Ensure: Final Clusters
Split D into n chunks: $D_1, D_2, \ldots, D_n$
Distribute each D_i to master nodes in distributed environment
for each D_i on master node S_i **in parallel do**
    Apply Firefly Algorithm to optimize D_i
    Store result as R_i
end for
Aggregate results $R_1, R_2, \ldots, R_n$ to slave node
Combine aggregated results into dataset R
Determine optimal K using Elbow Method on R
Apply K-means clustering on R using optimal K
repeat
    Assign every point to the closest centroid
    Revise centroids
until convergence
Output final clusters =0

Chapter 5

RESULTS AND DISCUSSIONS

In this section, the discussion is about the datasets used for the analysis, then followed by the setup required for the environment outlining the specific configurations and also followed by the output from the experiment with graphs and metrics table.

5.0.1 Experimental Setup

The setup included making a multi-node Apache Hadoop cluster on AWS EC2 utilizing t2.medium occurrences with Amazon Linux 2 AMI. Designed a VPC for secure organizing and connected 50 GB EBS volumes to each occasion. Hadoop 3.3.1 and OpenJDK 8 were introduced, and fundamental environment factors were set. The cluster comprised one ace hub running NameNode and ResourceManager administrations and numerous laborer hubs running DataNode and NodeManager administrations. We customized setup records (core-site.xml, hdfs-site.xml, yarn-site.xml, mapred-site.xml), set up passwordless SSH, designed the HDFS, and begun YARN administrations. The cluster's health and usefulness were confirmed utilizing Hadoop's web interfacing and command-line tools, ensuring a strong environment for the tests.

Table 5.1: Comparison of Clustering Approaches

Datasets	**Approach**	**2-Node Cluster**			**4-Node Cluster**		
		Purity Index	**DBI**	**Time (sec)**	**Purity Index**	**DBI**	**Time (sec)**
PokerHand	**Firefly Algorithm**	0.64	0.68	230	0.68	0.74	150
	K-Means	0.56	0.98	296	0.46	0.91	200
TabularPlayground	**Firefly Algorithm**	0.64	0.69	255	0.68	0.59	172
	K-Means	0.51	0.98	310	0.50	0.91	231

The Figure 5.1 and Figure 5.2 display the comparison of clustering performance metrics for 2-node clusters using the Firefly and Kmeans approaches on two datasets: PokerHand and Tabular Playground. The graph 5.1 presents the Davies-Bouldin Index (DBI), where lower values indicate better cluster separation. It reveals that for the PokerHand dataset, the Firefly method results in a better (lower) DBI compared to Kmeans, while for the Tabular Playground dataset, the Firefly approach shows a significantly better DBI than Kmeans. The

graph 5.2 illustrates the Purity Index, where higher values indicate better cluster quality. It shows that for the PokerHand dataset, the Firefly approach yields a higher purity index compared to Kmeans, while for the Tabular Playground dataset, the result is same. Overall, these graphs suggest that Firefly shows superior performance in terms of DBI and purity for both datasets.

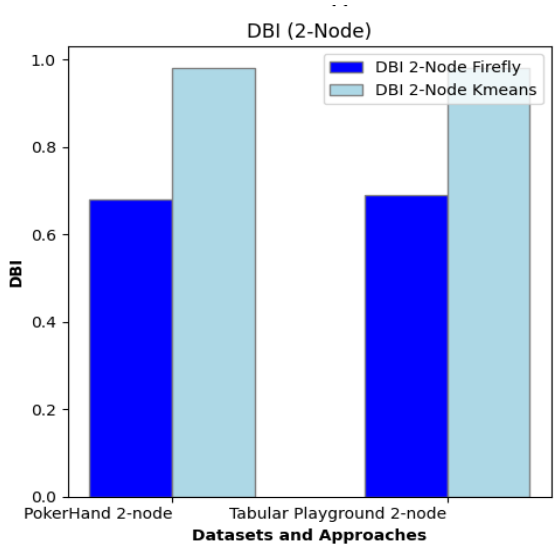

Figure 5.1: DBI for 2-node for Datasets

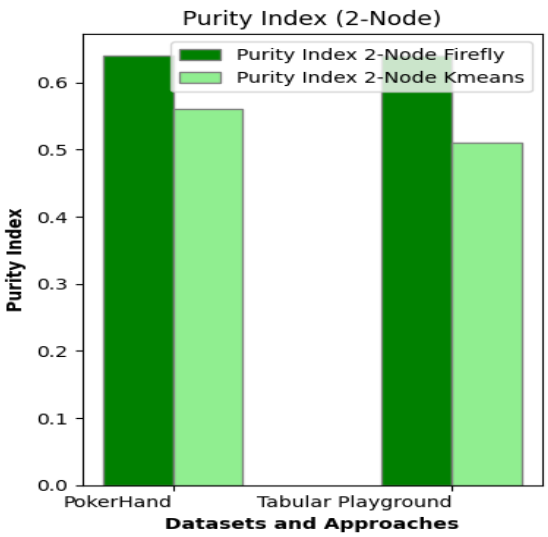

Figure 5.2: Purity for 2-node for Datasets

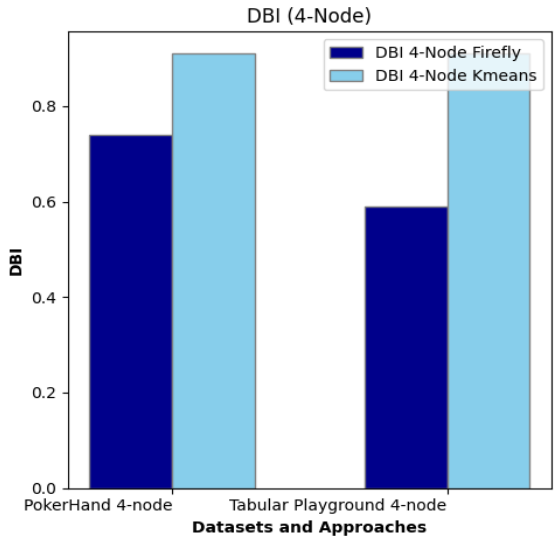

Figure 5.3: DBI for 4-node for Datasets

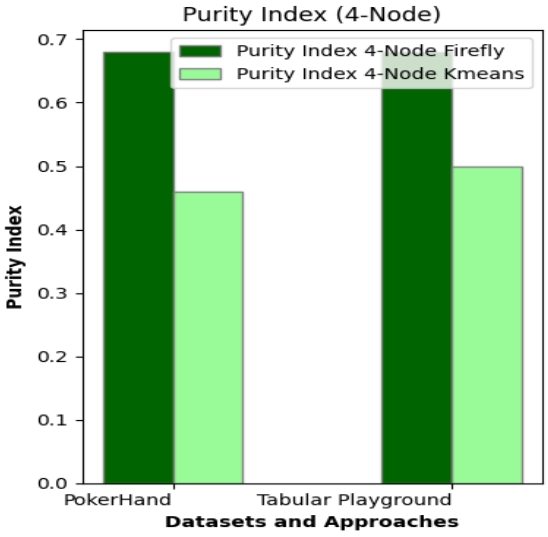

Figure 5.4: Purity for 4-node for Datasets

The Figure 5.3 and Figure 5.4 present a comparison between two clustering approaches, Firefly and K-means, on two datasets, PokerHand and Tabular Playground, for a 4-node setup. The graph 5.3 shows the Davies-Bouldin Index (DBI), where lower values indicate

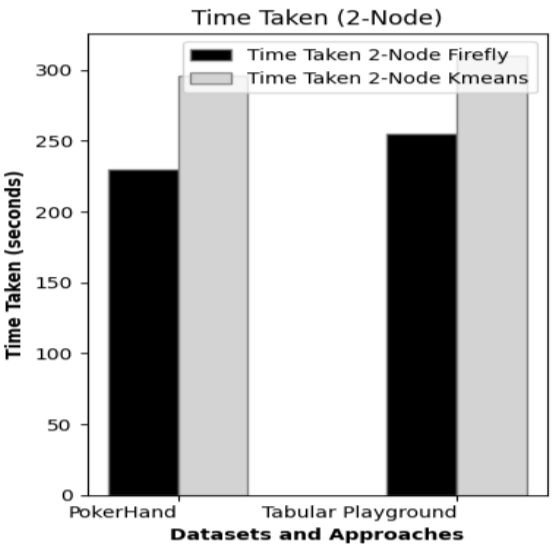

Figure 5.5: Time taken for 2-node setup

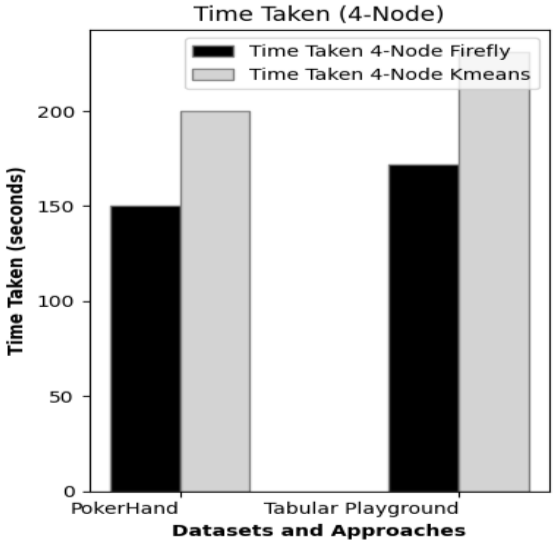

Figure 5.6: Time taken for 4-node setup

better clustering performance. For the PokerHand dataset, the DBI is lower for the Firefly approach compared to K-means, suggesting better performance by Firefly. Conversely, for the Tabular Playground dataset, Firefly exhibits a lower DBI than K-Means, indicating better clustering. The graph 5.4 depicts the Purity Index, where higher values signify better performance. For the PokerHand dataset, Firefly has a higher Purity Index, while for the Tabular Playground dataset also, Firefly achieves a higher Purity Index. These results indicate that Firefly performs better on the both datasets according to DBI and Purity Index.

From the Figure 5.5 of the 2-node setup, the Firefly Algorithm demonstrates a clear advantage in terms of time efficiency over K-Means for both datasets. Specifically, for the PokerHand dataset, the Firefly Algorithm takes 230 seconds, while K-Means takes about 296 seconds. Similarly, for the Tabular Playground dataset, the Firefly Algorithm requires around 255 seconds, whereas K-Means takes approximately 310 seconds.

In the Figure 5.6 of 4-node setup, the Firefly Algorithm continues to outperform K-Means in terms of time taken. For the PokerHand dataset, the Firefly Algorithm takes about 150 seconds compared to K-Means' 200 seconds. For the Tabular Playground dataset, the Firefly Algorithm requires roughly 172 seconds, while K-Means takes about 231 seconds. The time reduction is more significant for the Firefly Algorithm when moving from 2-node to 4-node setups, highlighting its better scalability and efficiency in a parallel processing environment.

Chapter 6

CONCLUSION AND FUTURE SCOPE

This research demonstrates the potential of integrating the Firefly Algorithm with traditional K-Means clustering to address the challenges posed by big data. Our experimental results indicate that the Firefly Algorithm outperforms K-Means in terms of clustering quality and efficiency. The Firefly Algorithm consistently shows lower Davies-Bouldin Index (DBI) and higher Purity Index across both 2-node and 4-node setups for the PokerHand and Tabular Playground datasets, indicating superior cluster separation and quality. Additionally, the Firefly Algorithm demonstrates significant time efficiency, taking less time than K-Means for both datasets and node configurations. This improvement in performance and efficiency is crucial in big data environments, where the ability to accurately and swiftly categorize vast and complex datasets can provide significant competitive advantages. The findings suggest that leveraging the Firefly Algorithm, either standalone or in a hybrid approach with K-Means, can enhance clustering performance, making it a valuable tool for big data applications.

The experiments conducted on the PokerHand and Tabular Playground datasets highlight the robustness and scalability of our approach. By employing the Firefly Algorithm for optimizing the number of clusters and using the Elbow Method for validation, we successfully addressed the limitations of conventional clustering techniques, such as sensitivity to initial conditions and difficulties in handling large datasets. The variability in clustering performance observed with our algorithm suggests further refinements and enhancing consistency.

D----1

by Minor P

Plagiarism Report

Project Title: Data Clustering using Firefly Algorithm in Distributed Environment

Team No: D1

Project Domain: Data Engineering

Submission date: 21-Jun-2024 04:23PM (UTC+0530)
Submission ID: 2406227085
File name: D1_repo_wrr.pdf (470.47K)
Word count: 4172
Character count: 24242

D----1

ORIGINALITY REPORT

16%	7%	12%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to B.V. B College of Engineering and Technology, Hubli Student Paper	5%
2	Lecture Notes in Computer Science, 2014. Publication	3%
3	Submitted to University of Nottingham Student Paper	1%
4	Mohammad Sultan Mahmud, Joshua Zhexue Huang, Rukhsana Ruby, Alladoumbaye Ngueilbaye, Kaishun Wu. "Approximate Clustering Ensemble Method for Big Data", IEEE Transactions on Big Data, 2023 Publication	1%
5	Aittokallio, T.. "Inspiratory flow shape clustering: An automated method to monitor upper airway performance during sleep", Computer Methods and Programs in Biomedicine, 200701 Publication	1%
6	www.slideshare.net Internet Source	1%

22	"Computational Intelligence in Data Mining - Volume 2", Springer Science and Business Media LLC, 2015 Publication	<1 %
23	"Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 2", Springer Science and Business Media LLC, 2018 Publication	<1 %
24	repository.ubaya.ac.id Internet Source	<1 %
25	"Multimedia Technology and Enhanced Learning", Springer Science and Business Media LLC, 2024 Publication	<1 %
26	Qiang Zhang, Hongxin Li, Changnian Liu, Wei Hu. "A New Extreme Learning Machine Optimized by Firefly Algorithm", 2013 Sixth International Symposium on Computational Intelligence and Design, 2013 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

REFERENCES

[1] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[2] Yuhui Shi et al. Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, volume 1, pages 81–86. IEEE, 2001.

[3] Mahamed GH Omran, Andries P Engelbrecht, and Ayed Salman. An overview of clustering methods. *Intelligent Data Analysis*, 11(6):583–605, 2007.

[4] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011.

[5] Xin-She Yang. Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*, 2(2):78–84, 2010.

[6] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[7] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[8] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[9] Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer, 2002.

[10] Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang. A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference*, pages 81–92. Elsevier, 2003.

[11] Ron Bekkerman, Ran El-Yaniv, and Andrew McCallum. Multi-way distributional clustering via pairwise interactions. In *Proceedings of the 22nd international conference on Machine learning*, pages 41–48, 2005.

[12] Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 328–339. SIAM, 2006.

[13] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):1–26, 2008.

[14] Kenneth Cukier. *Data, data everywhere: A special report on managing information*. Economist Newspaper, 2010.

[15] Inderjit S Dhillon and Dharmendra S Modha. A data-clustering algorithm on distributed memory multiprocessors. In *Large-scale parallel data mining*, pages 245–260. Springer, 2002.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[17] Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *Pattern Recognition*, 41(9):2742–2756, 2008.

[18] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *Proceedings of the 2nd ACM SIGOPS/EuroSys European conference on computer systems 2007*, pages 59–72, 2007.

[19] Makoto Iwayama and Takenobu Tokunaga. Cluster-based text categorization: a comparison of category search strategies. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 273–280, 1995.

[20] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39:459–471, 2007.

[21] Avita Katal, Mohammad Wazid, and Rayan H Goudar. Big data: issues, challenges, tools and good practices. In *2013 Sixth international conference on contemporary computing (IC3)*, pages 404–409. IEEE, 2013.

[22] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[23] Trupti M Kodinariya, Prashant R Makwana, et al. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[24] Christopher Moretti, Jared Bulosan, Douglas Thain, and Patrick J Flynn. All-pairs: An abstraction for data-intensive cloud computing. In *2008 IEEE international symposium on parallel and distributed processing*, pages 1–11. IEEE, 2008.

[25] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pages 42–47. IEEE, 2013.

[26] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[27] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

Appendix A

A.1 Data Visualization Details

1. The Figure 5.1 illustrates the Davies-Bouldin Index for a 2-node cluster computed on the PokerHand and TabularPlayground datasets. The index measures the average similarity between each cluster and its most similar cluster, with lower values indicating better clustering results.

2. The Figure 5.2 shows the purity value for a 2-node cluster on the same datasets. Purity measures the extent to which clusters contain predominantly a single class, with higher values indicating better clustering performance in terms of class separation.

3. The Figure 5.3 illustrates the Davies-Bouldin Index for a 4-node cluster computed on the PokerHand and TabularPlayground datasets. The index measures the average similarity between each cluster and its most similar cluster, with lower values indicating better clustering results.

4. The Figure 5.4 shows the purity value for a 4-node cluster on the same datasets. Purity measures the extent to which clusters contain predominantly a single class, with higher values indicating better clustering performance in terms of class separation.