

**WQD7007 DATA MINING**

**PROJECT REPORT**

**SEMESTER 2, SESSION 2023/2024**

**ASSIGNMENT: EVALUATE THE DIFFERENCES BETWEEN TIME- SERIES CLUSTERING AND DENSITY-BASED CLUSTERING IN BIG DATA ENVIRONMENT**

**Group: G2**

**Teacher: Tutut Herawan**

|  |  |
| --- | --- |
| **Name** | **Matric Number** |
| SAGAL MOHAMED YUSUF | 22093010 |
| TARSVINI A/P RAVINTHER | 17193844 |
| RABITA BHUIYA TANAYA | 22111473 |
| ZHANGXUAN | 22091244 |

Contents

[1. BASIC UNDERSTANDING 3](#_Toc1630415004)

[2. DATASET AND TOOL EVALUATION 5](#_Toc420141074)

[Describe the motivation to select the appropriate dataset in Big Data Environment. 5](#_Toc668276097)

[3. COMPARE RESULTS AND SUMMARY 9](#_Toc1373361615)

[REFERENCES 10](#_Toc1805132367)

# **1.** **BASIC UNDERSTANDING**

**Time-Series Clustering and Density-Based Clustering: An Overview**

A crucial part of studying temporal data in many domains is time-series clustering, which involves grouping time-series data according to their similarity. Two main types of techniques are employed in time-series clustering: distance-based methods and model-based methods. In contrast to distance-based approaches, which use metrics like Dynamic Time Warping (DTW) or Euclidean distance to calculate similarities or dissimilarities, model-based approaches fit a model to each time-series and group these models [1]. Significant difficulties in this area are caused by high dimensionality and temporal interdependence, which call for sophisticated methods like dimensionality reduction and feature extraction. Applications where it is critical to comprehend temporal patterns for decision-making include healthcare monitoring, stock market analysis, and climate trend detection. Time-series clustering plays a key role in these fields [2].

The primary goal of density-based clustering is to separate meaningful data points from irrelevant ones by locating dense zones and labeling them as such. In this class, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the most famous algorithm. In densely populated regions, DBSCAN clusters points, whereas in sparsely populated areas, it identifies individual points as outliers [3]. Because it is resistant to outliers and noise, this approach is great for finding clusters of any size or shape. In contrast to k-means, which necessitates the pre-specification of the cluster number, DBSCAN defines density based on characteristics such as the neighborhood radius and the minimum number of points needed to create a cluster. Anomaly detection, picture segmentation, and geographical data analysis are some of the many applications of this technique in geographic information systems [4].

Both time-series clustering and density-based clustering are invaluable in data mining. Time-series clustering enables the analysis of temporal patterns, which is crucial for forecasting and anomaly detection, while density-based clustering excels in identifying complex and irregularly shaped clusters in noisy data, enhancing insights in spatial and image data analysis.

**Importance and Applications of Time-Series and Density-Based Clustering in Data Mining:**

Sequences across time for data miners to successfully discover and analyze trends and patterns across many domains and time periods, clustering is an essential tool. It plays a crucial role in the financial sector by classifying stock price movements into similar categories, which helps with risk assessment and portfolio management [5]. By keeping tabs on patients' vital signs, time-series clustering aids in the early diagnosis of health problems [6]. To improve load distribution and estimate demand, it clusters consumption patterns in energy management [7]. Improving operational efficiency and strategic planning, time-series clustering allows for the extraction of significant patterns from temporal data, which in turn aids decision-making and predictive analytics.

Based on Density In order to differentiate between dense and sparse areas in spatial data, clustering is essential for revealing intricate patterns. Identifying high-density areas, like urban development zones, and spatial anomalies, like environmental hotspots, are two of the most common uses of this technique in geographic information systems (GIS) [8]. One use of density-based clustering in cybersecurity is the detection of suspicious network activity [9]. In addition, it facilitates the segmentation of consumer data according to buying habits, which allows for more focused marketing techniques in the retail sector [10]. One effective method for delving into complicated datasets is density-based clustering, which is noise-tolerant and can find clusters of any shape.

The capacity to derive profound insights from large and varied datasets is what makes these clustering algorithms so important in data mining. Density-based clustering makes it easy to investigate and comprehend spatial and irregular data patterns, but time-series clustering lets one predict future trends and behaviors. Inspiring innovation and enhancing decision-making, their applications cover a wide range of industries.

**Challenges in Applying Time-Series and Density-Based Clustering Techniques in Big Data Environments:**

**Time-Series Clustering in Big Data:**

There are several major obstacles to implementing time-series clustering in a big data setting. The increased computational complexity and memory utilization that might result from time-series data's large dimensionality is one such issue. When faced with massive amounts of time-series data, traditional clustering algorithms may not be able to handle them efficiently, leading to sluggish performance or possibly clustering failure [1].

Clustering is further complicated by the presence of noise, missing values, and abnormalities in time-series data. Improving, normalizing, and smoothing are crucial preprocessing stages; yet, when working with large datasets, they can be computationally demanding [6]. Furthermore, complex distance measures such as Dynamic Time Warping (DTW) are computationally costly and not readily scalable to big data when it comes to precisely aligning time-series data and capturing temporal interdependence [2].

The diversity of time-series data sources, each with its own unique sampling scale, length, and pace of data collection, adds another layer of complexity. To further complicate matters, this heterogeneity calls for state-of-the-art methods of feature extraction and dimensionality reduction to establish a consistent foundation for clustering [11].

**Density-Based Clustering in Big Data:**

DBSCAN and other density-based clustering methods encounter unique difficulties when applied to massive data settings. Scalability is a major concern with these algorithms. The computational complexity of DBSCAN increases dramatically as the dataset size increases, rendering it unfit for big datasets unless certain adjustments are made [3]. As the amount of data points grows, the algorithm's requirement for pairwise distance computations can become unaffordable.

Another difficulty is the sensitivity of the parameters. To obtain the best clustering results, you need to fine-tune DBSCAN's settings, such as the neighborhood radius (epsilon) and the minimal number of points (MinPts). Suboptimal clustering or high processing costs can result from incorrect parameter selection in a big data setting, which is exacerbated by the data's sheer volume and variety [4].

Another challenge for density-based clustering is dealing with data that has a lot of dimensions. Due to the sparse distribution of data points in high-dimensional spaces, significant dense regions are difficult to spot. Dimensionality reduction and feature selection are necessary but computationally intensive techniques [12].

# **2.** **DATASET AND TOOL EVALUATION**

**Describe the motivation to select the appropriate dataset in Big Data Environment**

In a Big Data context, choosing the right dataset is essential for many reasons, all of which improve the efficacy and efficiency of data analysis and decision-making procedures. There are several reasons for this choice's motivation:

**1. Pertinence to Business Goals**

- Alignment with Goals: The dataset must align with the specific business goals or research questions. Irrelevant data can lead to misleading insights and poor decision-making.

- Actionable Insights: The right dataset helps in deriving actionable insights that can directly influence business strategies and outcomes.

**2. Data Quality and Integrity**

- Accuracy and Reliability: Ensuring that the data is accurate and reliable is fundamental. Poor quality data can result in erroneous conclusions.

- Completeness: A comprehensive dataset that includes all necessary variables and observations is essential for thorough analysis.

**3. Volume and Scalability**

- Handling Large Volumes: The dataset should be manageable within the existing technological infrastructure. Datasets that are too large might require significant processing power and storage, leading to higher costs.

- Scalability: The dataset should be scalable to accommodate future data growth without requiring a complete overhaul of the data architecture.

**4. Variety and Diversity of Data**

- Multi-Source Integration: Incorporating data from diverse sources (structured, semi-structured, and unstructured) can provide a more comprehensive view and richer insights.

- Coverage: Ensuring the dataset covers all necessary dimensions and variables relevant to the analysis.

**5. Timeliness and Recency**

- Up-to-Date Information: In many scenarios, especially in real-time analytics, the dataset needs to be current to provide relevant insights.

- Historical Data: For trend analysis and forecasting, historical data is essential. Balancing between current and historical data is crucial.

**6. Data Privacy and Security**

- Compliance with Regulations: The dataset must comply with relevant data privacy laws and regulations (e.g., GDPR, HIPAA). Non-compliance can lead to legal repercussions.

- Data Security: Ensuring the dataset is secure from unauthorized access and breaches is vital.

**7. Cost-Effectiveness**

- Budget Constraints: The dataset's acquisition, archiving, and processing costs ought to be within the organization's financial constraints.

- Return on Investment (ROI): The costs associated with the dataset should be justified by the possible advantages and insights it offers.

**8. Technical Compatibility**

- Integration with Existing Systems: The dataset needs to work with the current tools, technologies, and data infrastructure.

- Ease of Processing: The format of the data should allow for easy processing and analysis with the knowledge and resources already at hand.

**9. Analytical Value**

- Depth of Insights: Predictive modeling and machine learning, among other advanced analyses, should be possible with the dataset's depth of data.

- Support for Advanced Analytics: It ought to facilitate a range of analytical methods, including data mining, machine learning, and statistical analysis.

We chose the weather dataset [13] as it is a streaming dataset, which collects the weather details over a period.

DataSource: <https://www.kaggle.com/datasets/muthuj7/weather-dataset/data?select=weatherHistory.csv>

Figures 1,2,3 and 4 show the WEKA results for data clustering.

Figure 1: **-** Simple K-Means

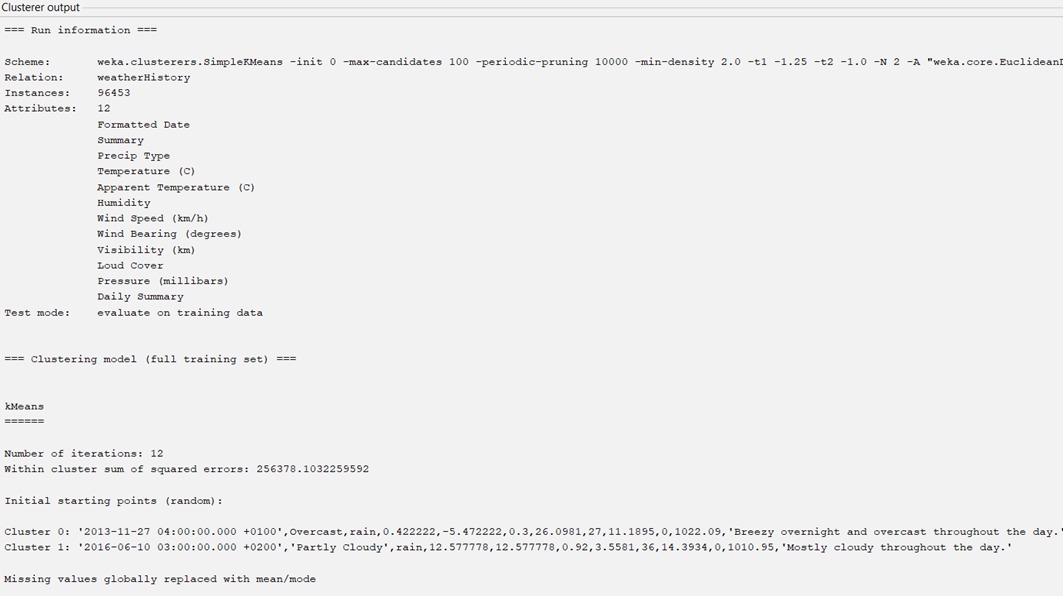


Figure 2: **-** Simple K-Means

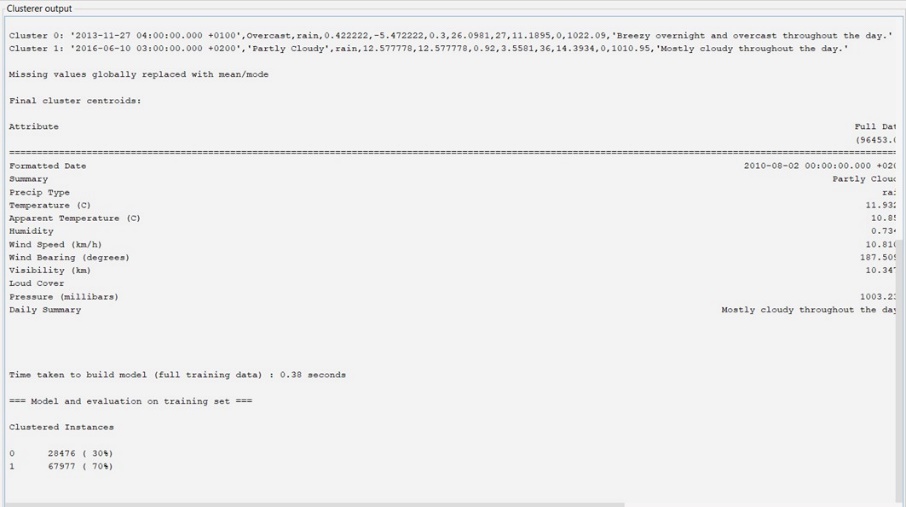


Figure 3: **-** DBSCAN Results

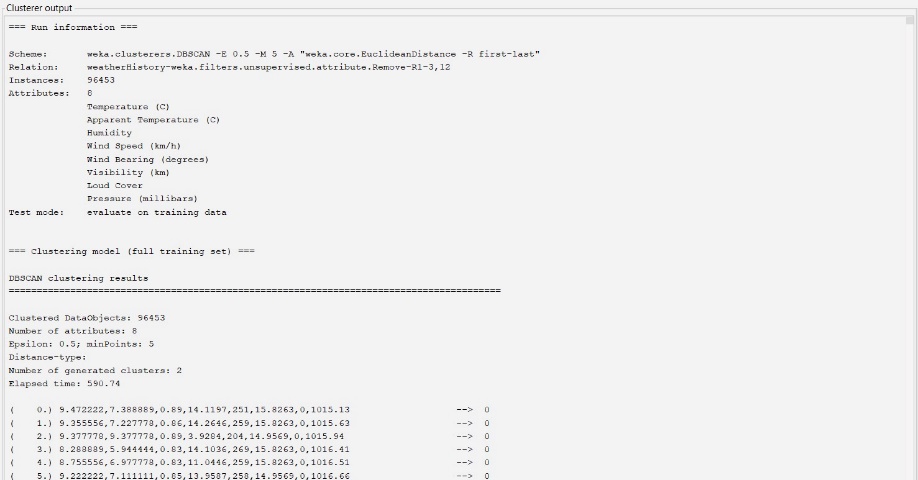
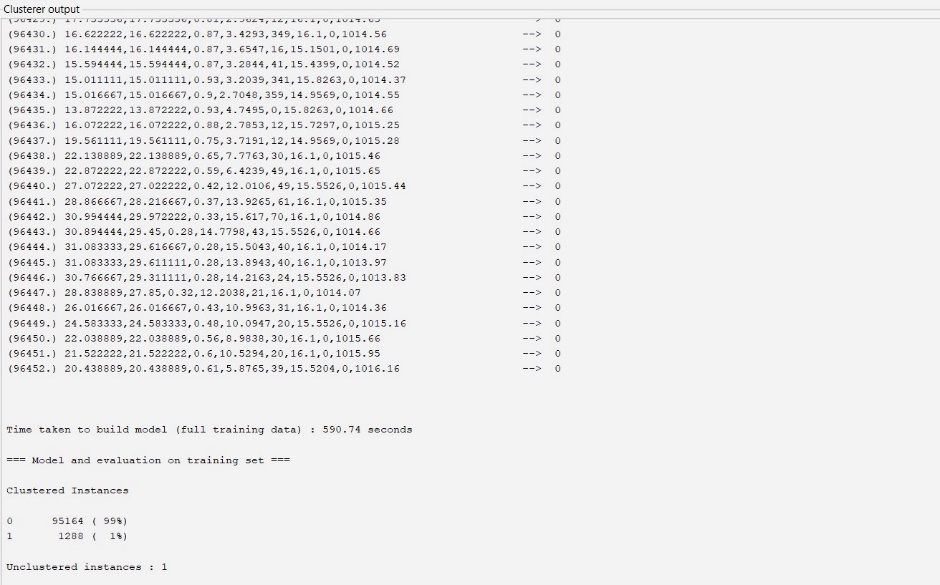


Figure 4: **-** DBSCAN Results



**Choice of Tool: WEKA**

**Reasons for Choosing WEKA:**

This setup ensures that you effectively handle large datasets with potential noise and complex cluster shapes, leveraging WEKA's user-friendly interface and powerful data mining capabilities.

1. User-Friendly Interface: WEKA provides a GUI that simplifies the process of applying machine learning algorithms to datasets.

2. Comprehensive Toolkit: WEKA includes a wide range of algorithms for clustering, classification, and other data mining tasks.

3. Visualization Capabilities: WEKA offers excellent data visualization options, which can help in analyzing the results of the clustering.

4. Support for Time-Series Data: WEKA has specific filters and preprocessing tools for time-series data, making it easier to handle and analyze such datasets.

**Reasons for choosing Simple K-Means:**

K-means clustering is a popular and widely used technique for partitioning a dataset into K clusters, where each data point belongs to the cluster with the nearest mean. Here are some reasons why k-means clustering might be chosen, particularly for large datasets:

**1. Simplicity and Efficiency**

- Easy to Implement: K-means is straightforward to understand and implement. Its algorithm involves simple mathematical computations which make it easy to use.

- Computational Efficiency: K-means is computationally efficient, with a time complexity.

**2. Scalability**

- Handles Large Datasets: K-means can handle large datasets effectively, given that its computational complexity grows linearly with the number of data points. This makes it well-suited for big data applications.

**3. Interpretability**

- Intuitive Results: The results of k-means clustering are easy to interpret. The centroids represent the average position of all the points in the cluster, providing a clear and understandable summary of the data structure.

**4. Flexibility**

- Applicable to Various Data Types: K-means can be applied to a variety of data types, including continuous, categorical (with appropriate pre-processing), and mixed data types.

- daptable: Variations of the basic k-means algorithm (such as k-medoids, k-means++) can address specific requirements and data characteristics.

**5. Suitability for High Dimensional Data**

- Dimensionality Reduction: K-means can be combined with dimensionality reduction techniques (like PCA) to handle high-dimensional data efficiently. This combination can improve the performance and accuracy of clustering.

**Reasons for Choosing DBSCAN:**

1. Handling Noise: Resilient to outliers is DBSCAN. In contrast to k-means or k-medoids, which are susceptible to noise, DBSCAN can locate and isolate noise locations.
2. Non-Spherical Clusters: DBSCAN's ability to identify clusters with any shape is advantageous since real-world data frequently deviates from spherical distributions.
3. No Need to Specify Number of Clusters: DBSCAN uses the density distribution of the data to identify the number of clusters, as opposed to k-means or k-medoids, which require you to define the number of clusters in advance.

# **3. COMPARE RESULTS AND SUMMARY**

### **Comparison of K-Means and DBSCAN Clustering Techniques:**

### The K-Means algorithm divided the dataset into two clusters, with one cluster containing 30% of the instances and the other 70%. This distribution indicates that the data has two main patterns, with one being more prevalent. The algorithm aims to minimize the variance within each cluster, making it suitable for datasets with well-defined, spherical clusters. The computation was efficient, taking only 0.38 seconds to build the model. However, K-Means may not handle noise well and can be sensitive to the initial placement of centroids.

### DBSCAN identified two clusters, with 99% of the instances in one cluster and 1% in another, along with one unclustered instance. This suggests that most of the data points form a dense region, while a small portion represents outliers or less dense areas. DBSCAN does not require specifying the number of clusters beforehand and can find arbitrarily shaped clusters. It is robust to noise and outliers but computationally intensive, taking 590.74 seconds to build the model. DBSCAN's effectiveness depends on the choice of parameters, such as the neighborhood radius and the minimum number of points required to form a cluster.

### **Detailed Explanation:**

### The K-Means clustering results show that the dataset is split into two distinct groups, with a clear majority pattern. The within-cluster sum of squared errors indicates how tightly the data points are grouped around the centroids. This method is straightforward and computationally efficient but may struggle with non-spherical clusters and noise.

### In contrast, DBSCAN's results highlight a dominant dense cluster and a few outliers. This method excels at identifying clusters of varying shapes and sizes and is effective in handling noise. The small second cluster and unclustered instance suggests that DBSCAN can separate dense regions from sparse ones, providing insights into the data's density distribution.

### **Summary and Potential Future Applications:**

### Both K-Means and DBSCAN offer valuable insights into clustering in big data environments. K-Means is suitable for scenarios where clusters are well-defined and spherical, providing quick and efficient clustering. DBSCAN is ideal for datasets with complex shapes and noise, identifying dense regions and outliers effectively.

### Potential future applications for these techniques include:

### K-Means: Customer segmentation, market basket analysis, and image compression.

### DBSCAN: Anomaly detection in network security, spatial data analysis in GIS, and identifying patterns in large-scale time-series data.

# **REFERENCES**

[1] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” in *Proc. International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1578-1585, doi: 10.1109/IJCNN.2017.7966039.

[2] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Adversarial attacks on deep neural networks for time series classification: A review,” *Data Mining and Knowledge Discovery*, vol. 35, no. 3, pp. 1131-1171, 2021.

[3] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN revisited, revisited: why and how you should (still) use DBSCAN,” *ACM Transactions on Database Systems (TODS)*, vol. 46, no. 3, pp. 1-21, 2021.

[4] Y. Zhao, L. Xu, and G. Chen, “An enhanced DBSCAN algorithm with parameter optimization and its application to spatial clustering in GIS,” *Applied Sciences*, vol. 10, no. 6, p. 2158, 2020.

[5] Y. Wu, Y. Wang, Z. Cai, and H. Wang, “Time-series clustering for stock recommendation,” *Expert Systems with Applications*, vol. 161, p. 113696, 2020.

[6] M. Malik, A. A. Khan, and S. Ali, “Time-series clustering for health monitoring: A deep learning approach,” *IEEE Access*, vol. 9, pp. 14765-14777, 2021.

[7] H. Ponce, P. Ponce, A. Molina, and F. Rodriguez, “Time series clustering for demand forecasting in smart grid applications,” *Energies*, vol. 13, no. 7, p. 1718, 2020.

[8] C. Deng, S. Zhang, and J. Yang, “Density-based clustering algorithm with noise for large spatial databases,” *Information Sciences*, vol. 526, pp. 64-82, 2020.

[9] S. Shah, B. Issac, and S. Jacob, “An efficient density-based clustering approach for anomaly detection in smart grid systems,” *Sustainable Cities and Society*, vol. 61, p. 102365, 2020.

[10] C. Chen, Q. Zhao, and S. Yu, “Customer segmentation in e-commerce using density-based clustering,” *Electronic Commerce Research and Applications*, vol. 46, p. 101019, 2021.

[11] Y. Yang and M. L. Shyu, “Heterogeneous Time Series Clustering for Human Action Recognition,” *IEEE Transactions on Multimedia*, vol. 22, no. 2, pp. 362-374, 2020.

[12] T. T. Tran, C. Luo, and D. Phung, “Fast and Efficient Density-Based Clustering for Large-Scale Time Series Data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1166-1180, 2020.

[13] Muthukumar.J, "Weather Dataset," *Kaggle.com*, 2017. <https://www.kaggle.com/datasets/muthuj7/weather-dataset/data?select=weatherHistory.csv> (accessed Jun. 20, 2024).

‌