Python-based Graphical User Interface for Automatic Selection of Data Clustering Algorithm

Mohamed A. Amashaa[1], Dalia khairya[2], Rania A. Abougalalaa[3],Salem Alkhalaf[4] ,Marwa F.Areed[5]

1,2,3 Department of Computer Teacher Preparation, Damietta University, Damietta, 34511, Egypt

4Faculty of Science and Arts, Computer Science Department, Qassim University, Alrass, Saudi Arabia.

5Faculty of Engineering, department of computer science, Damietta University, Damietta,34517, Egypt.

[1]mw_amasha@yahoo.com , [2]shamaamora2014@gmail.com , [3]Ronyabogalala@hotmail.com , [4]s.alkhalaf@qu.edu.sa , [5]marwa_areed@du.edu.eg

Abstract

The process of grouping a set of similar data objects in the same group based on similarity criteria is called clustering. There are many clustering algorithms and software tools. Currently, K-means and Weka are the most common clustering algorithms and tools, respectively. The Weka tool does not contain all possible clustering algorithms and does not provide a comparative study between them to illustrate the differences and the suitable one for the dataset used. In this context, the main purpose of this paper is developing highly interactive graphical software application to make a comparative analysis of nine different clustering algorithms, including the K-means algorithm, Mean Shift algorithm, Affinity Propagation, and Density-based algorithm to choose the compatible one in terms of efficiency and accuracy. The simulation is done by a graphical user interface (GUI) software system designed by Python on a general data set. The limitations and directions for future research are also presented.

Keywords: *We would like to encourage you to list your keywords in this section*

1. Introduction

Data mining technology (DMT) is the process of integrating traditional methods of data analysis with complex algorithms in order to extract accurate, useful information from an enormous amount of unused data, which may later be used to predict an event in the future. DMT has evolved into a developing innovation to separate valuable examples and data from enormous informational indexes [1]. The application of DMT brings a new dimension to support decision prediction [2].

Data mining is widely used as an indicator of decision-making in many areas, such as education, trade, medical data, sports, and politics. Furthermore, it is used to analyze huge amounts of data and summarize it in the form of useful information that can be used in decision-making and to reduce costs. It is frequently adopted as an answer to computational problems in different research areas and has become influential in many areas[3].

The development of information and communications technology (ICT) has created huge amounts of information from various sources, which might be kept in a variety of areas. Every database can use its own system to save information [4]. There are many clustering algorithms and software tools such as the K-means clustering algorithm of Weka, which are the most common clustering algorithms and tools. The main objective of this research is to make a comparative analysis of nine different clustering algorithms, including the K-means algorithm, Mean Shift algorithm, Affinity Propagation algorithm, and Density-based algorithm. These algorithms are compared in terms of efficiency and accuracy. The current paper is organized as follows: Section 2 presents a theoretical background to uncover the current limitations and research gaps; Section 3 presents the system design and architecture in addition to discussing the simulator designed for this study; Section 4 presents the results and discussion; and lastly, Section 5 concludes this research and suggests future work.

2. Theoretical Background

Clustering is a machine learning technique that involves the grouping of data points. Given a set of data points, a clustering algorithm can be used to classify each data point into a specific group. This section gives a brief description of nine clustering algorithms.

2.1 K-means Algorithm

The K-means clustering algorithm is a partition-based cluster separation technique. According to the algorithm, we prime pick target number k as the primary cluster centroids, then measure the length among any object and any cluster center and attach it to the nearest center, get the averages of all clusters, and replicate this method till the criterion function has converged [5].

2.2 Mini Batch K-means Algorithm

The Mini Batch K-means is an alternative of the K-means algorithm, which applies mini-batches to decrease the calculation time, while trying to optimize for purposes of similarity. Mini-batches are subsets of the figures data, randomly tested in any training repetition [6].

2.3 Mean-Shift Algorithm

The mean-shift algorithm is a strong and adaptive clustering algorithm with non-parametric density estimation, and it does not need prior information about the number of clusters [6].

2.4 Affinity Propagation Algorithm

Affinity Propagation (AP) was studied by Frey and Dueck, who described it as a powerful clustering methodology that propagates messages of affinities between pairwise points in a factor graph [7]. Compared to traditional approaches, the AP technique can also use nonmetric similarities as input data, making the data analysis exploration suitable for unusual metrics of similarity [8,9].

2.5 Spectral Algorithm

Spectral clustering is a technique known to perform well particularly in the case of non-gaussian clusters where the most common clustering algorithms such as K-Means fail to give good results. However, it needs to be given the expected number of clusters and a parameter for the similarity threshold [10].

2.6 Agglomerative Algorithm

Agglomerative hierarchical clustering begins by working with all entities in the form of primary clusters. In the first step, two of the entities are blended and the algorithm ends by producing one large cluster [11].

2.7 DBSCAN Algorithm

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm. In this method, clustering is based on density, such as density connected points. The DBSCAN method shows clusters as sections of huge density separated by blocks of low density [12].

2.8 BIRCH Algorithm

The balanced iterative reducing and clustering using hierarchies (BIRCH) method has been improved particularly for big datasets, particularly if the complete data cannot be stored in memory [13].

2.9 Gaussian Mixture Clustering Algorithm

The Gaussian mixture method (GMM) is a statistical model addressing a data population as a mix of multivariate customary (Gaussian) distributions [14].

2.10 Clustering Methods

These clustering algorithms have peculiar features and have been broadly categorized based on these into three categories: hierarchical methods, partitioning methods, and density-based methods. Partitioning clustering algorithms are aimed at determining k clusters that optimize distance-based or any other criteria [15]. On the other hand, hierarchical algorithms create a hierarchical decomposition database that can be presented in the form of a dendrogram. As far as density-based algorithms are concerned, they search for dense regions within the data space that are separated from each other by low density noise regions. The table below summarizes a comparative study of various algorithms under a number of methods by taking into consideration various aspects of clustering. Table 1 shows a summary of the pros and cons of each algorithm [16,17].

Table 1: Pros and Cons of Clustering Methods

Clustering method	Clustering Algorithms	Pros	Cons
Hierarchical	• BIRCH Algorithm • Spectral Algorithm • Agglomerative Algorithm	• Embedded flexibility based on the granularity level .• Appropriate for problems that involve point linkages such as taxonomy trees .• Applicable to any attribute types	• Inability to make corrections after making the splitting/merging decision • Lacking interpretability with regard to cluster descriptors • Vague termination criterion • For massive high dimensional

Clustering method	Clustering Algorithms	Pros	Cons
			datasets, it is prohibitively expensive
Partitioning	• K-means Algorithm • Mini Batch K-means Algorithm	• Relatively simple and scalable • Appropriate for datasets having well-separated compact spherical clusters	• Poor cluster descriptors • Degradation in high dimensional spaces • Highly sensitive to initialization phase, outliers, and noise
Density based	• DBSCAN Algorithm • Mean Shift Algorithm • Affinity Propagation Algorithm • Gaussian Mixture Clustering Algorithm	• Discovery of arbitrary-shaped clusters having varying sizes • Noise and outliers resistant	• Poor cluster descriptors • Highly sensitive to the input parameters' setting • Not suitable for high-dimensional datasets

3. Simulation

In this section, the introduced simulator is implemented using the Python programming language (Python 3.7 with PyCharm edition 2017.2.3). The proposed simulator can apply any of the nine clustering algorithms to any specific data type easily and show the number of clusters and the total processing time. But the main purpose of the introduced simulator is to make a comparative study between the nine clustering algorithms and showing the comparative result as a function in time, which is used as a performance metric. Finally, the simulator acts as a decision support system that identifies the best clustering algorithm for a specific data file.

In Figure 1, the process of the simulator is summarized while in Figures 2, 3, and 4, two options have the same processing scenario, starting with choosing the data file, determining the number of clusters (optional and has a default value of 3) and the independent variable index, which starts with index 0 for the first column (does not exist in the Comparing Algorithms Screen) and then applying the chosen algorithm or algorithms. Even though the two options have the same scenario, the results for the two cases (options) are different. In the first case (option one), the output is the drawing of the clustered data and the total processing time is calculated while the output in the second case (option two) is a comparison of applying the chosen algorithms using seven different parameters

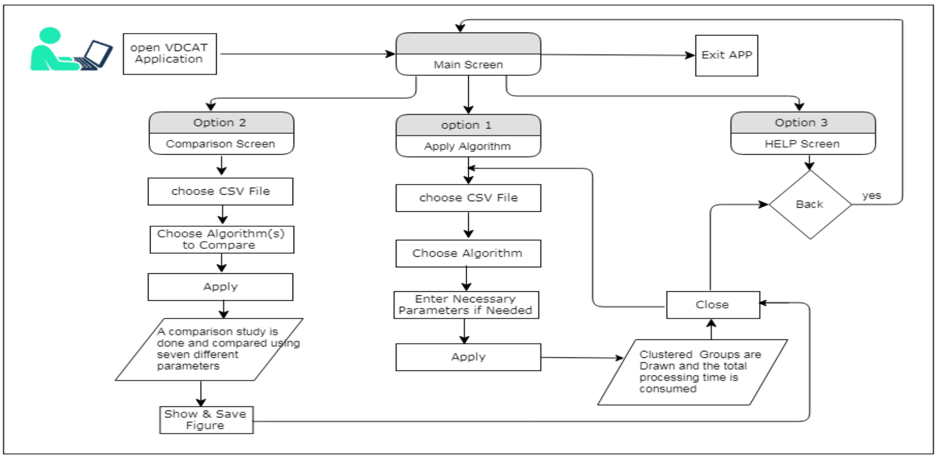
open VDCAT Application
Main Screen
Exit APP
Option 2
Comparison Screen
option 1
Apply Algorithm
Option 3
HELP Screen
choose CSV File
Choose Algorithm(s) to Compare
Apply
A comparison study is done and compared using seven different parameters
Show & Save Figure
choose CSV File
Choose Algorithm
Enter Necessary Parameters if Needed
Apply
Clustered Groups are Drawn and the total processing time is consumed
Close
Back
yes

Figure 1: The simulator process

3.1 Comparison Metrics

The seven metrics can be summarized in brief as follows [18]:

- **TP:** Time Processing score. This determines the total processing time of applying such an algorithm on any data file; a low value is better.
- **H:** Homogeneity score. This score is useful to check whether the clustering algorithm meets an important requirement: a cluster should contain only samples belonging to a single class. It is bounded between 0 and 1, with low values indicating a low homogeneity.
- **C:** Completeness score. The purpose of this score is to provide an item of information about the assignment of samples belonging to the same class. More precisely, a good clustering algorithm should assign all samples with the same true label to the same cluster. It is bounded between 0 and 1; high values are better.
- **VM:** V-measure score. This is computed as the harmonic mean of the distinct homogeneity and completeness scores, just as this score is the harmonic mean between homogeneity and completeness [19].
- **AR:** Adjusted Rand. This score is defined as the number of pairs of objects that are either in the same group or in different groups in both partitions divided by the total number of pairs of objects. The Rand index lies between 0 and 1; a high value is better.
- **AMI**: The Adjusted Mutual Info score is used to compare clusters. It measures the similarity between the data points that are in the clustering, accounting for chance groupings and takes a maximum value of 1 when clustering are equivalent [20].
- **S:** Silhouette score. This score measures how similar an object is to its own cluster compared to other clusters. The silhouette scores range from -1 to 1, where a higher value indicates that the object is better matched to its own cluster

and worse matched to neighboring clusters. If many points have a high value, the clustering configuration is good.

3.2 Simulator Main Screens

The Developed simulator has five screens in addition to the result screens. The main screen has four options. As shown in Figure 2.

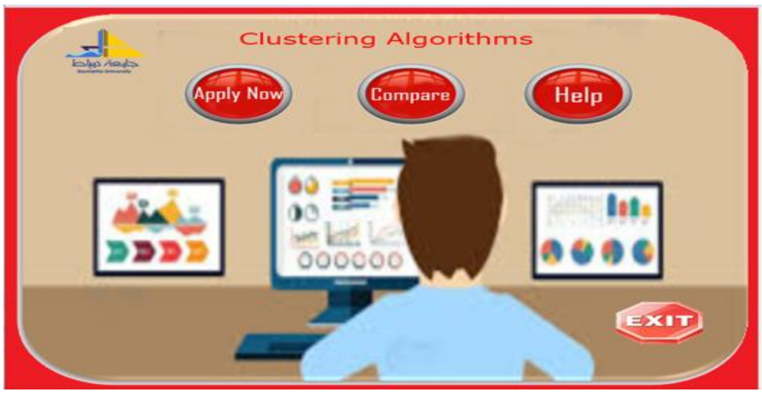

Figure 2: Main Screen of the Developed Software Tool

The first option is “Apply Now”: In this option, you can apply any clustering algorithm on any data file “CSV files”. After that, the processing time is consumed and clusters are drawn. As shown in Figure 3.

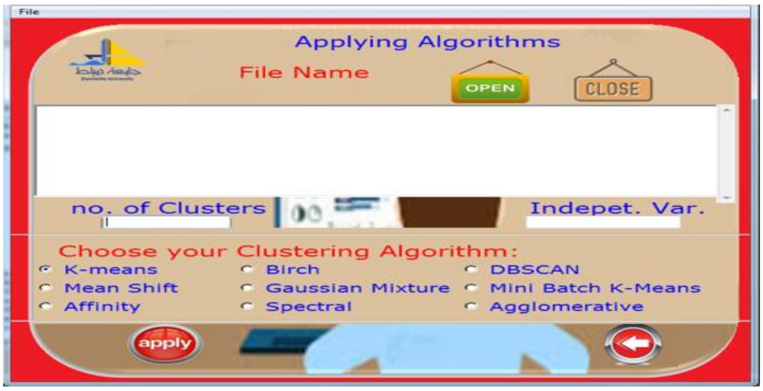

Figure 3: Algorithm Applying Screen

In the second option, “Comparison,” it is permissible to perform a comparative study between any algorithms of the possible nine. This includes even between all of them on any chosen data files and gives a full comparison using seven different metrics before giving a decision of the best clustering algorithm. The Algorithms Comparison options screen is shown in Figure 4.

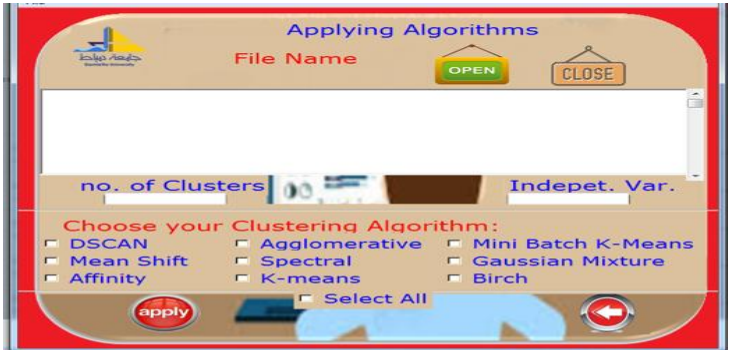

Figure 4: Algorithms Comparison Screen

4. Experimental Results and Discussion

In this section, a brief evaluation and discussion is introduced to show and illustrate the simulator screen. The simulator is applied on different datasets that differ in file size or data types (numbers or letters) or number of tuples. Table 2 summarizes the used datasets specifications.

Table 2: Used Data Files Specification

	File name	Size	No. Col	No. of tuples
1	Gasoline	1 k-byte	4	40
2	data_multivar	2 k-byte	2	100
3	Iris	4 k-byte	5	150
4	Shopping data	5 k-byte	5	200
5	Sample stocks	5 k-byte	2	649
6	dividendinfo-1	7 k-byte	6	200
7	Internet	29 k-byte	6	963

In this section, a series of controlled experiments were conducted using the simulator in the seven different files defined above in Table 2. The experiments are done for the comparison option between clustering algorithms when applied to a specific file.

4.1 Applying Different nine Clustering Algorithms

This section consists of browsing a comparison between different applied algorithms on a specific CSV data file using seven different parameters, as explained in section 3.1. As seen in Figure 5, a comparison between the nine clustering algorithms on File 1 tells us that agglomerative clustering gives the best total processing time while mini-batch and K-means are the best in homogeneity and similarity.

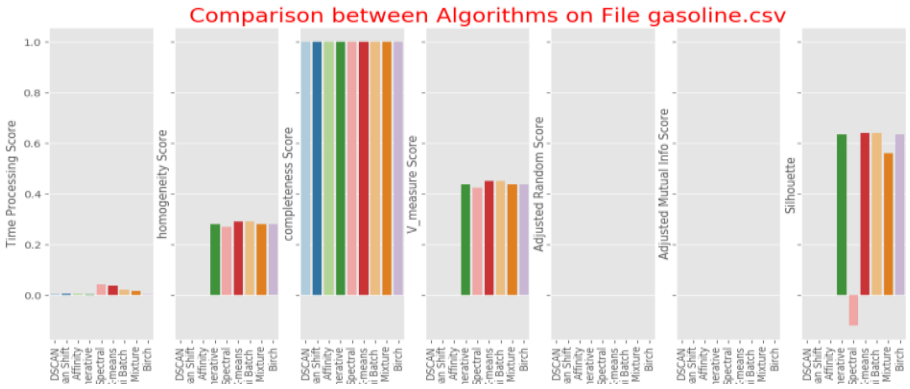

Figure 5: Comparative Study on File 1

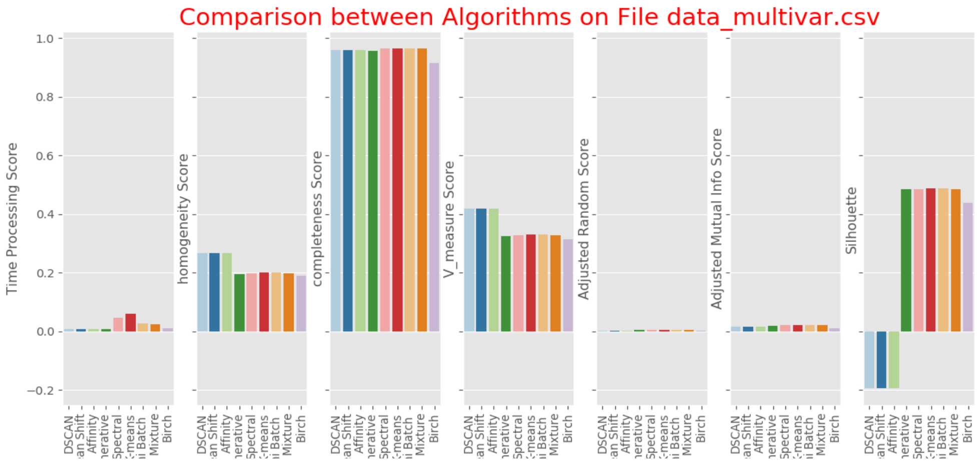

Figure 6: Comparative Study on File 2

Also, Figure 6 shows that BIRCH or DBSCAN clustering algorithms give the best total processing times while the Mean Shift Algorithm, DBSCAN, and Affinity Propagation are the best in homogeneity and completeness.

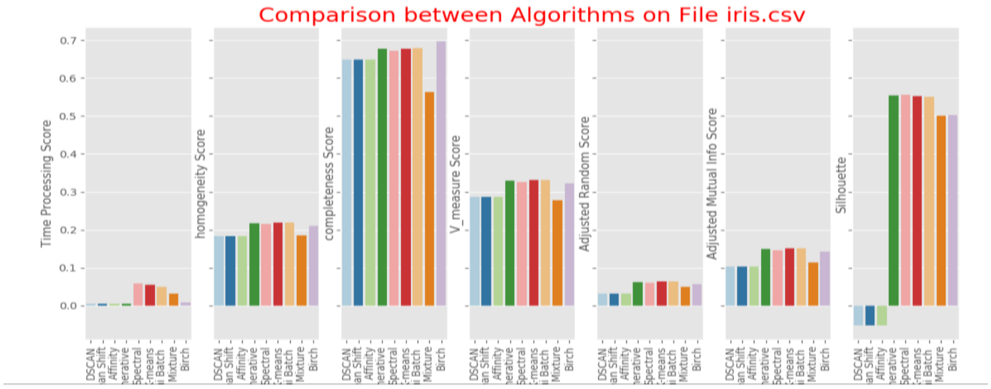

Figure 7: Comparative Study on File 3

Figure 7 shows that the DBSCAN clustering algorithm gives the best total processing time, but Mini-Batch and Gaussian Mixture are the best in homogeneity, completeness, and similarity.

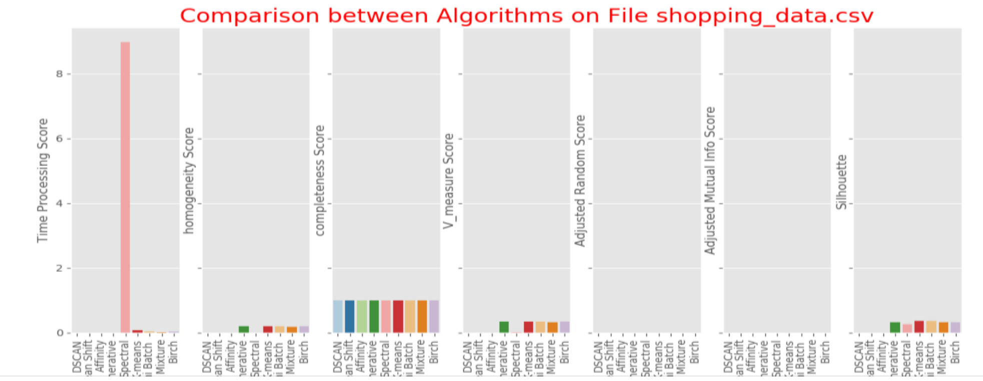

Figure 8: Comparative Study on File 4

Figure 8 shows that the DBSCAN clustering algorithm gives the best total processing time, while K-means is the best in homogeneity, completeness, and similarity. But in Figure 9, the DBSCAN clustering algorithm gives the best total processing time while DBSCAN, Mean Shift, and Affinity Propagation are the best in completeness and similarity.

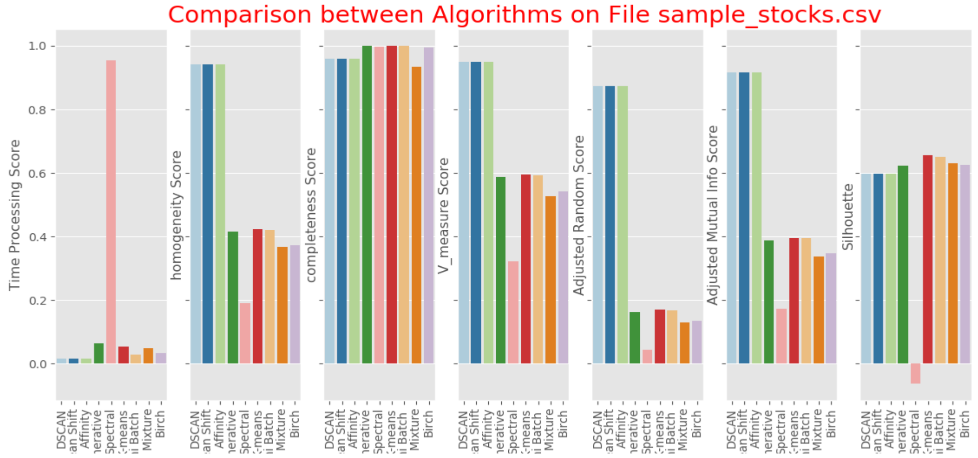

Figure 9: Comparative Study on File 5

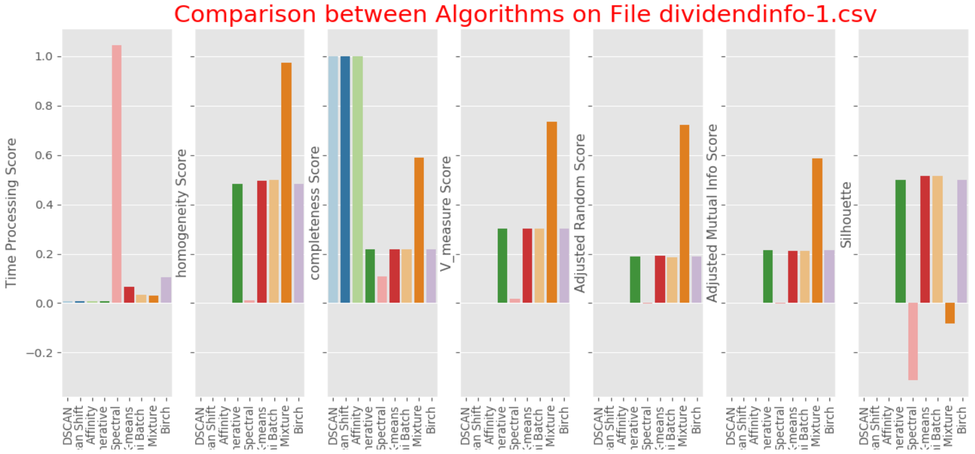

Figure 10: Comparative Study on File 6

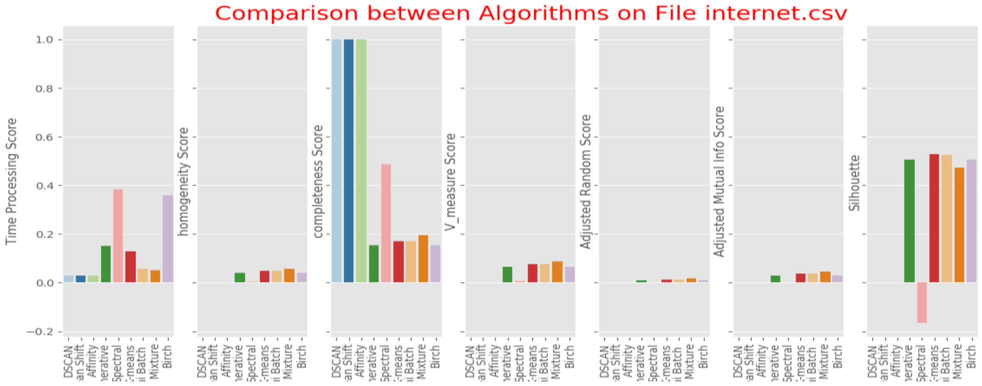

Figure 11: Comparative Study on File 7

From the simulation and results, it is concluded that when applying the nine different clustering algorithms on any comma-separated values file (CSV file), the best one from the point of view of the seven metrics is determined, as mentioned before in section 3. As seen in the resulting figures, some cases can use all metrics to differentiate between algorithms, and in other cases, processing time is the only metric that can be used to produce a decision. In the implementation of the developed tool, numerous programming problems arose; they were solved, one following another, and the tool was developed and finished. Also, in the experiments, the tested files were limited in size according to the used computer specifications because computer resources must increase as data file size increases.

Based on the results of this research, our study can contribute to the literature in an interdisciplinary field, by using the simulator to get good results for analyzing models based on large datasets for the behavior of materials.

5. Conclusion and Future Work

Weka is one of the more common tools used to apply clustering algorithms on a dataset. It includes K-means, which is one of the more common clustering algorithms, but it does not contain the other algorithms mentioned in this work, so we cannot conduct comparative studies between any of the algorithms. In this context, it w Clustering Algorithm Decision taking system as necessary to find a way to make a comparison study between the different algorithms using different comparison metrics. Currently, a GUI simulator designed by Python is one of the most powerful and frequently used programming languages. It can be applied to any general dataset.

As a future work, this research can be extended by completing the simulation of other clustering algorithms and applying the simulator on a large number of datasets in both amount and size.

Abbreviations

AMI - Adjusted Mutual Info
AR - Adjusted Rand
C - Completeness score
CSC - Compressive spectral clustering
CSV - comma-separated values file

DMT - Data mining technology
GMM - Gaussian mixture method
GUI - *Graphical user interface*
ICT **-** Information and communications technologies

6. References

[1] Shi, G.-r. and X.-S. Yang. “Optimization and data mining for fracture prediction in geosciences.” Procedia Computer Science 1(1) (2010): 1359–1366.

[2] Shadroo, S. and A. M. Rahmani. “Systematic survey of big data and data mining in internet of things.” Computer Networks 139 (2018): 19–47.

[3] Jha, J. and L. Ragha. “Educational data mining using improved apriori algorithm.” International Journal of Information and Computation Technology 3(5) (2013): 411–418.

[4] Wang, R., et al. “Review on mining data from multiple data sources.” Pattern Recognition Letters 109 (2018): 120–128.

[5] Wang, J. and X. Su. An improved K-Means clustering algorithm. 2011 IEEE 3rd International Conference on Communication Software and Networks, IEEE (2011).

[6] Yang, G., et al. "Remote sensing of seasonal variability of fractional vegetation cover and its object-based spatial pattern analysis over mountain areas." ISPRS Journal of Photogrammetry and Remote Sensing 77 (2013): 79–93.

[7] Frey, B. J. and D. Dueck. "Clustering by passing messages between data points." Science 315(5814) (2007): 972–976.

[8] Guan, R., et al. "Text clustering with seeds affinity propagation." IEEE Transactions on Knowledge and Data Engineering 23(4) (2010): 627–637.

[9] Moiane, A. F. and Á. M. L. Machado. "EVALUATION OF THE CLUSTERING PERFORMANCE OF AFFINITY PROPAGATION ALGORITHM CONSIDERING THE INFLUENCE OF PREFERENCE PARAMETER AND DAMPING FACTOR." Boletim de Ciências Geodésicas 24(4) (2018): 426–441.

[10] Ramasamy, D. and U. Madhow. Compressive spectral embedding: sidestepping the SVD. Advances in Neural Information Processing Systems (2015).

[11] Chong, C. Y., et al. "Efficient software clustering technique using an adaptive and preventive dendrogram cutting approach." Information and Software Technology 55(11) (2013): 1994–2012.

[12] Raviya, K. H. and K. Dhinoja. "An Empirical Comparison of K-Means and DBSCAN Clustering Algorithm." PARIPEX Indian Journal of Research 2(4) (2013): 153–155.

[13] Fontanini, A. D. and J. Abreu. A Data-Driven BIRCH Clustering Method for Extracting Typical Load Profiles for Big Data. IEEE Power & Energy Society General Meeting (PESGM), IEEE (2018).

[14] Göhring, A., et al.. "Using Gaussian Mixture Model clustering for multi-isotope analysis of archaeological fish bones for palaeobiodiversity studies." Rapid Communications in Mass Spectrometry 30(11) (2016): 1349–1360.

[15] Kameshwaran, K. and K. Malarvizhi. "Survey on clustering techniques in data mining." International Journal of Computer Science and Information Technologies 5(2) (2014): 2272–2276.

[16] Baser, P. and J. R. Saini. "A comparative analysis of various clustering techniques used for very large datasets." International Journal of Computer Science & Communication Networks 3(5) (2013): 271.

[17] Gondaliya, B. "Review Paper on Clustering Techniques." International Journal of Engineering Technology, Management and Applied Sciences 2(7) (2014).

[18] Garreta, R., et al. scikit-learn: Machine Learning Simplified: Implement scikit-learn into every step of the data science pipeline, Packt Publishing Ltd. (2017).

[19] Rosenberg, A. and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL) (2007).

[20] Yeung, K. Y. and W. L. Ruzzo. "Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data." Bioinformatics 17(9) (2001): 763–774.