EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Clustering based Contact Tracing Analysis and Prediction of SARS-CoV-2 Infections

Meenu Gupta[1], Rakesh Kumar[1], Sunil Kumar Chawla[1,*], Sunny Mishra[1] and Sourabh Dhiman[1]

[1]Computer Science and Engineering, University Institute of Engineering, Chandigarh University, Mohali, Punjab, India

Abstract

INTRODUCTION: Contact tracing is a method to track the victims, which have been infected from the host with any particular disease. Therefore, clustering based machine learning techniques can be employed for contact tracing. Contact tracing can be automated by using technology and thus helps us in producing much more accurate and efficient results.

OBJECTIVES: This work aims at finding usefulness of clustering techniques for contact tracing.
Two different clustering techniques namely density-based clustering and partitioning-based clustering have been used to analyse corresponding results for COVID-19 infected cases. The dataset is generated from a mock data generator with certain assumptions.

RESULTS: The paper compares DBSCAN and K-means for contact tracing for COVID-19 Pandemic. The comparative analysis of two algorithms is presented.

CONCLUSION: The effectiveness of certain clustering algorithms in COVID-19 contact tracing is analysed. DBSCAN performs well for clustering tasks. This work only focuses on possible techniques useful for contact tracing and does not claim any medical accuracy.

Keywords: Clustering algorithm, Contact tracing, DBSCAN, SARS-CoV-2, COVID-19

Received on 28 May 2021, accepted on 31 October 2021, published on 03 November 2021

doi: 10.4108/eai.3-11-2021.171756

*Corresponding Author. Email:drskchawla.1983@gmail.com

1. Introduction

The effectiveness of modern technologies such as Artificial Intelligence (AI), Big Data, Internet of Things (IoT), Machine Learning, and Deep Learning [18] in medical sciences and health care has increased at an astounding rate in the past few years [1]. This study intends to analyse and evaluate the effectiveness of various machine learning clustering algorithms in COVID-19 contact tracing. The Coronavirus disease (COVID-19) pandemic has disrupted the economy and lives of people on a global scale. COVID-19 is much more communicable than other viruses like SARS and MERS. But the mortality rate of this virus is lower than the other viruses. Initially, there are very mild or no symptoms for the COVID-19 carrier. The carrier had already spread the virus to the others meeting the carrier before he was tested positive for the virus. This shows us the great need for the 'contact tracing' to find the discreet persons who met COVID-19 virus carrier.

There are multiple ways to put a check the spread of this virus. One of the ways is the pharmaceutical way, which uses vaccination or antibiotics/ anti-viral drugs. Another way is to prohibit people from infecting each other. This can be achieved either by reducing the hygienically related contacts or to trace the contacts of infected individuals. Contact tracing is done either

manually or digitally. In manual contact tracing, health authorities interview the infected individual. The prime agenda of the interview is to identify those individuals who came in close contact with the carrier individual for more than 5-10 minutes in the last 14-21 days [6]. On this basis of context (e.g. indoors/outdoors), proximity (distance of contact between two individuals), duration of contact these health authorities then compute a risk score for each individual coming in contact. However, this is not possible for the carrier to remember all the individuals who met them in the last 14-21 days. They might also have infected unknown individuals, for example, contact with an individual in the supermarket. Also, this chain continues for each infected individual. Moreover, manual contact tracing requires a lot of time and workforce of health authorities.

Therefore, Researchers are aiming to build a technological solution to automate this process via digital contact tracing. The goal of this digital contact tracing is to find rapidly the contacts that are at great risk to be infected. This kind of contact tracing is strongly based on the networks of contact of the people linking together. These contacts can be linked together to form a cluster and from these clusters, we can identify the infected individuals. We can use machine learning techniques to build these clusters. Various types of clustering algorithms are available. These clustering algorithms are of different types like Centroid-based Clustering, Density-based Clustering, Distribution based Clustering, Hierarchical clustering, Partitioning based clustering. In this study, we are focusing on Density-based clustering and partitioning-based clustering.

Density-based clustering is a type of unsupervised learning. This technique utilizes the idea that a cluster is a high point density contiguous region in the data space separated by low point density contiguous region clusters. The data points which are there in the separating regions of low point density are termed as outliers/noise [7-8]. Partitioning-based data clustering is a type of unsupervised learning. This provides us a refined and abstract view of the data structure by partitioning it into a specific number of disjoint groups [9].

Different architectures can be considered to collect the data for tracing purposes. These are the centralized, decentralized, and hybrid approaches [10]. These architectures use the Blue trace protocol where the requirement is that the user should be pre-registered on some central server. This generates a temporary id for each device nearby using Bluetooth technology. These encounters can be recorded and utilized as an input for the clustering algorithms. Tracing coverage and testing delays play a major role in contact tracing. Considering the ideal case having 0 days delay and covering 100% of tracing, we can omit the spread between the individuals up to a significant level.

2. Literature Survey

In [3] the author proposed an extensive review of different clustering techniques present in the data mining. This works provides a piece of detailed information regarding different types of clusters like well-separated clusters, center-based clusters, contiguous clusters, contiguous clusters, Density-based clusters, and shared property/conceptual clusters. They also mentioned the properties that a good clustering algorithm should have. Different classifications of the clustering have also been discussed like Hierarchical methods, partitioning methods, and Density-based methods. The author concluded that any clustering "is a division of the objects into a group based on a set of rules – it is neither true nor false."

In [17], the author proposed a Density-Based algorithm that can discover the clusters in the large spatial databases in presence of noise. The algorithm is named DBSCAN. This algorithm fulfills the requirement of clustering in larger spatial databases i.e. minimum knowledge to determine the input parameters and the ability to discover clusters efficiently in large databases with arbitrary shapes. This algorithm relies on the density-based notion of cluster. This algorithm requires only one input parameter. It also supports the user to determine the appropriate value of this input parameter. For the effectiveness and efficiency evaluation of this algorithm, the author used synthetic data and real data of SEQUOIA 2000 bench-mark. The author concluded that the DBSCAN defeated the CLARANS algorithm by a factor greater than 100 in terms of efficiency. Hence, the DBSCAN algorithm is more effective than the CLARANS algorithm in discovering clusters of arbitrary shapes.

In [2] the author introduced a modelling study on the impacts of time intervals on the potency of various contact tracing methods. The projected mathematical model evaluates the impact of time intervals and completeness in various steps involved in tracing strategies. These delays include the interval of the time between getting infected, identifying symptoms, and various testing delays. This proposed model also includes the tracing of close as well as casual contacts. They have also computed the effectual multiplicative numbers of contact tracing approaches with a group of people having distance matrix, different scenarios for the isolation index, and segregation of their contacts. Author concluded that for testing and testing delays of 0 days and tracing coverage of 100%, the effective reproduction number of 1.2 comes down to 0.8 by adding contact tracing. Also, the author concluded that the proportion of onward transmission per index case that can be prohibited depends on the interval of testing and tracing, an interval of 0 days between tracing and testing from up to 79.9% compared to the delay interval of 3 days as 41.8% and 4.9 with a 7-day testing delay.

In [6] author implies a comprehensive review of COVID-19 contact tracing applications in terms of their system design, privacy, Proximity estimation, security, and information management. This work provided

detailed information on different architectures like centralized, Decentralized, and Hybrid. They also provided detailed information regarding the different steps followed in these different architectures. This work also provides details regarding how this different contact-tracing application collects the data and utilizes them in contact tracing and privacy issues related to the data collected by these applications. Different contact tracing applications are analyzed on different protocols specific to privacy, Security, proximity estimation, and data management. The author concluded his work by giving directions for the research of upgraded design of applications on the parameters of tracing and security potency.

3. Materials and Methods

The geographical spread of the different people is shown in Figure 1. The fuzzy Logic System comprises four chief divisions: fuzzifier, rule-base, inference engine, and a de-fuzzifier. Initially, the marks in various attributes are provided as a crisp input to the model. These inputs are then transformed into a fuzzy collection using fuzzy linguistic terms and membership functions. Afterward, they are passed through an IF-ELSE rule base. As a third move, the outcome of fuzzified output is charted to a crisp output by applying the membership functions, in the defuzzification move.

3.1 Dataset Description

The data set of this work has been collected from an online mock data generator [15], which contains geographical location data of different individuals. This dataset is meant to test only one of many possible scenarios on contact tracing models; it doesn't however claim to cover all possible cases and variations that exist in the real world. The generated dataset contains 100 total location tags of 10 different individuals, 12 UNIX timestamps have also been supplied with a gap of 60 minutes (i.e. 3600 seconds) to mimic the real-world movement of people over an area [16]. Every record of the dataset contains its corresponding unique identifier column called *id*, this is only meant to distinguish between records of the same individuals. Each record is in the form of a JSON object.

3.2 Pre-Processing Steps

The dataset used for this study must abide by certain criterion to be used effectively during model training:

3.2.1 Population Density

Since the important factor in the transmission of this virus is contact and transmission through air, so closer the distance between people greater are the chances of spreading the virus. Therefore, the population density plays a vital role in performance algorithms on the dataset. If the geographical locations of individuals are far apart or greater than a certain minimum transmission distance, the actual propagation of contagion may not take place.

3.2.2 Time Constraint

The physical presence of any two individuals not only should be within 6 feet of each other but the time at which they come in contact with each other should also be the same, otherwise, virus transmission would not take place. In the real world, people would commute from one place to another and in doing so come in contact with multiple people and thus greatly increasing the probability of contagion infecting people in their vicinity. Hence, the dataset must have a balanced number of records with similar timestamp values.

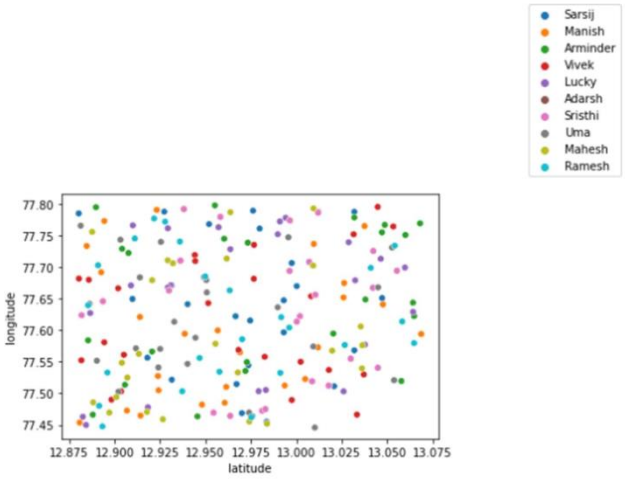

Figure 1. The geographical spread of different people within the data set

3.2.3 Number of instances of each individual

The number of records of each individual must be more than one in number to encompass the movement of an infected individual between multiple locations; this would provide an accurate simulation for the real-life movement of people. In this work, the mock dataset generated contains 10 instances of each differentiable by timestamp and geographic location tags.

3.3 Model Formulation

Clustering deals with the partition of different data elements or entities and combines those forming groups based on similarity in characteristics called clusters. Clusters are entities that share similar characteristics among themselves whilst dissimilar characteristics with objects of other cluster [3]. It's a data analysis technique used in data science to identify or discover certain patterns or characteristics of data and group them accordingly for gaining insights about the data. Representation of data using fewer clusters leads to

inevitable data loss or loss of certain finer details but it also leads to simplification and overall generalization [3]. Clustering is an unsupervised machine learning technique. It is a machine learning technique in which user supervision is not necessary. The data need not be labelled and the model discovers patterns on its own and clusters/groups them accordingly. For the abovementioned reasons clustering algorithms are not only utilized for data organization and data categorization rather for data compression applications and model construction. By extracting similarities in the data, it is relatively less complex to represent the data using lesser symbols or notations [4].

Clustering techniques generate groups or clusters that should suffice the following criteria: Homogeneity; instances that belong to the same cluster are similar to each other and Intra-cluster non-homogeneity; each cluster varies from other clusters in terms of their objective characteristics. The clusters thus generated may have different characteristics [3-4]: Exclusive; any instance belongs to only one group. Overlapping; any instance may belong to multiple clusters, Probabilistic; an instance may have a certain probability of belongingness to a cluster. In this study, we have implemented two different clustering techniques:

3.3.1 Density-based Clustering:

Summation of the density functions of every object is termed as the overall density of the data points. Clusters thus formed determined by parameters called density attractors, where the aforementioned local maxima of the overall density function are termed as density attractors. The influence, however, may be arbitrary [3]. Clusters are formed based on the density of the region- examples are DBSCAN (Density-Based Spatial Clustering of Application with Noise) and OPTICS (Ordering Points to Identify Clustering Structure). DBSCAN is a widely used clustering algorithm. It does not require cluster numbers as a parameter; it infers the cluster numbers according to the data. The clusters generated by it can take arbitrary shapes; the clusters generated by K-means usually take spherical shape. Figure 2 is representing non-linear separable clusters whereas figure 3 shows linear separable clusters.

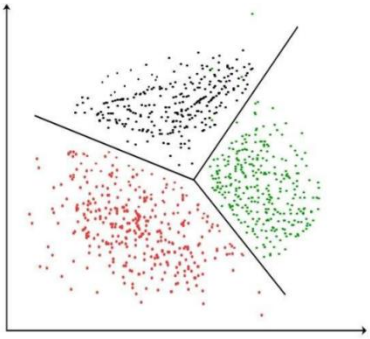

Figure 2. Non-Linear Separable clusters [3]

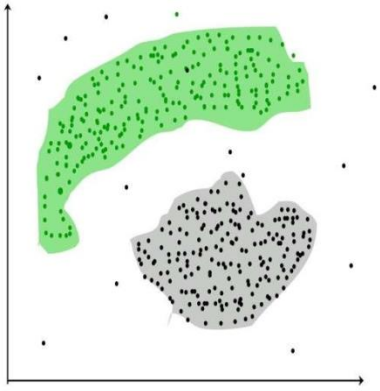

Figure 3. Linear-Separable Clusters [3]

3.3.2 Partitioning-based Clustering:

The data is partitioned into k units and each unit is termed as a cluster- examples are K-means, CLARANS (Clustering Large Applications based upon Randomized Search) as shown in figure 4. K-means begins with selecting k numbers of centroids (K is a hyper-parameter selected by the user depending upon the data characteristics, Model specification, and requirements) and iteratively readjusting the positions of centroids. When there is a relatively minute change, in the position of the centroids, the iteration is terminated and final clusters are obtained. The distance metric is chosen depending upon the data – examples are Euclidean distance, Manhattan distance, Hamming distance, etc.

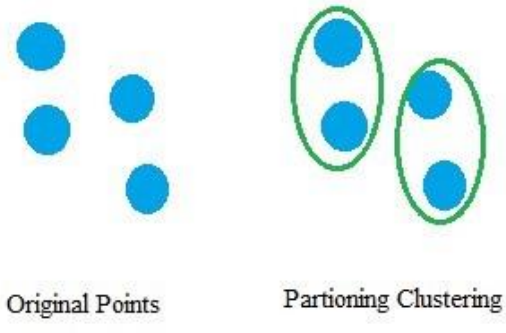

Figure 4. Data points selected by partitioning clustering for cluster formation [4]

3.3.3 Mathematical Analysis:

A set of n vectors $X_j, j = 1, \ldots, n$ have to partition into c groups $G_i, i = 1, \ldots, c$. The cost function, which can be based on Euclidean distance, Manhattan Distance or Hamming Distance between any vector X_k in group j and the corresponding cluster c_i, can be defined by [4]. Where J_i is the cost function within the group i.

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k, x_i \in G^i} \left\| x_i - c^i \right\|^2 \right) \quad (1)$$

K-Means algorithm requires a parameter k that defines the number of clusters. There are two most commonly used methods of selecting the optimal number of clusters from K-Means; Elbow Method and Silhouette Methods. The basic idea behind elbow methods is testing every number k and evaluating the value of cost function for each value of k and plotting a graph for the same.

$$0 < k < n \; ; \; where \; n = total \; number \; of \; data \; points - i$$

The graph begins with a high-cost value and gradually flattens; we select the cluster number where the change in slope of growth depicts the highest flattening also called the elbow of the graph. The second method is called Silhouette Method. In this method, we progress further by assuming that the data points have already been grouped into k number of clusters. Then for each data point, we define the following:

- $G(i) =$ *The cluster assigned to the ith data point*
- $|G(i)| =$ *The number of data points in the cluster assigned to the ith data point*
- $a(i)$: *gives a measure of how well assigned the ith data point is to its cluster*

$$a(i) = \left(\frac{1}{|G_{(i)}|} - 1\right) \sum_{G_{(i)}, i \neq j} d(i,j) \rightarrow ii \quad (2)$$

- $b(i)$: *gives the average dissimilarity to the closest cluster which is not its cluster*

$$b(i) = min_{i \neq j} \left((1/|G_{(j)}|) \sum_{j \in G_j} d(i,j) \right) \rightarrow iii \quad (3)$$

The silhouette coefficient $s(i)$ is given by:

$$s(i) = (b(i) - a(i)) / (\max (a(i), b(i)) \rightarrow iii \quad (4)$$

Silhouette is determined for every value of k and k having the maximum value of Silhouette coefficient determines the optimum number of clusters for the K-Means.

We have used different approaches to generate clusters from the dataset i.e. K-Means Clustering and DBSCAN. The number of clusters generated would eventually depend upon the dataset, In DBSCAN however, the number of clusters is 29; (cluster-0 to cluster-28). The significant arguments of the DBSCAN model that operates upon are listed below:

eps: provides maximum distance between two data points to be considered as neighbors of one another; in this work, eps represents the minimum safe distance as directed by the medical authorities and/or government to prevent the transmission of coronavirus

min_samples: the number of data points in the vicinity for a point to be considered as a core point

metric: distance metric to evaluate similarity among data points; here we use have sine metric that evaluates the distance on the three-dimensional Cartesian plane which is necessary for evaluating the distance between geographical locations on earth's

Similarly, then we generate clusters using K-Means Clustering. It requires certain critical cluster numbers as the argument to the model. The optimum cluster numbers can be decided by the elbow method and exploratory data analysis. Once the cluster formation is complete, tracing the infected individual is simple. We assume that any person *'person_a'* is infected, the model will simply find the cluster with *'person_a'* with associated 'id' from the dataset and append the *'person_a'* along with other entities (people) in the cluster into the infected list. After this step, each unique name from the list is extracted and the previous step is re-iterated until no more infected people are found then the iteration terminates. From the aforementioned steps, the formulated model generates a list of infected individuals.

4. Experimental Results and Discussion

There are various clustering algorithms and therefore various implementations. The performance algorithms depend upon the programming language utilized by the programmers to implement them, the underlying data structures, the computing node, and the dataset. An implementation in C and C++ will surely result in higher performance than the ones implemented purely in python. However, the underlying data structures can greatly affect the overall asymptotic performance of the algorithm. Similarly, the dataset also plays a pivotal role in the performance of algorithms. The size and statistical distribution of the dataset can have different effects in different algorithms. Hence, below we perform a comparative study of multiple clustering algorithms and datasets of varying sizes, where each algorithm is executed on a battery of randomly generated datasets of different sizes to accurately capture the worst-case performance on a broader spectrum. Fig. 5 and Fig. 6 depict two of the many dataset sub-types utilized to test the algorithms. The dataset depicted in Fig. 6 below has been reduced in dimensions, features, and sample size for the sake of formatting purposes in this paper [12]. The different clustering algorithm implementations being tested are mentioned below:

1. K-Means clustering (Sklearn and Scipy)
2. DBSCAN clustering
3. Agglomerative clustering (Sklearn and Scipy)
4. Spectral clustering

5. Affinity Propagation
6. Fastcluster
7. DeBaCl
8. HDBSCAN

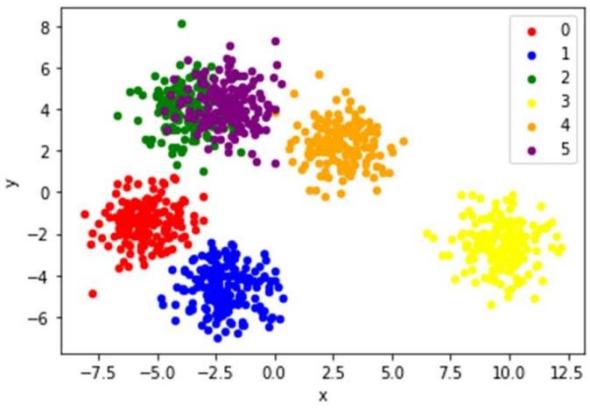

Figure 5. Dataset Sub-Type 1 (Blobs)

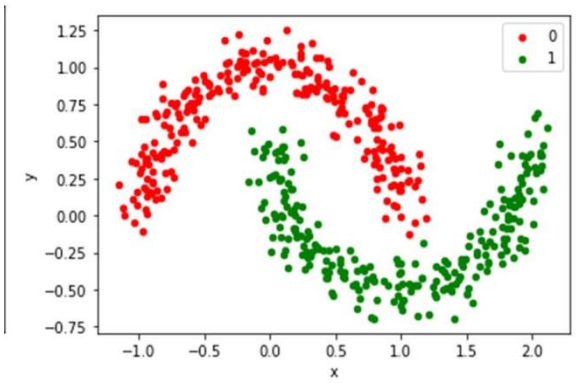

Figure 6. Dataset Sub-Type 2 (Moon)

During testing runs, ten different clustering algorithms were run many times over twenty-two datasets of varying sizes. If these algorithms were to run completely then they would be requiring hours of run-time on a workstation of standard configuration, hence run-time was cropped to 45 seconds per run for relatively slower algorithms. The comparison of all the above implementations of clustering algorithms is done by fitting a regression function to obtain a general scaling trend. We use the $seaborn$ library that provides $regplot$ which supports higher-order regression through the dataset. The trend lines or regression lines corresponding to each algorithm include a point to mark mean and one error bar which covers the range of data [12].

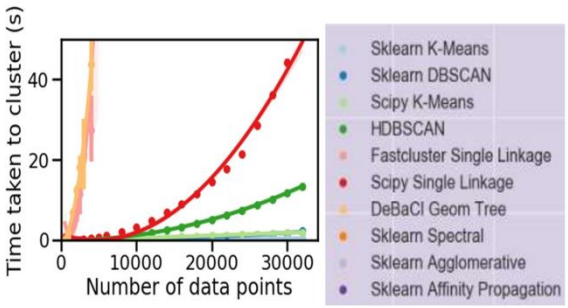

Figure 7. Clustering algorithm trend lines

From Fig.7 we can observe the faster algorithms were K-Means, HDBSCAN, single linkage agglomerative clustering, fast cluster single linkage, and DBSCAN. The important point to remember is that this is merely an attempt to gain a rough insight into the asymptotic performance of various clustering algorithms. Hence, validation of generated clusters is not necessary [12]. After analysis, we can surmise that DBSCAN along with other algorithms performed significantly better than the likes of DeBaCl Geom Tree, Scipy Spectral, and Sklearn Affinity Propagation algorithm. We have chosen DBSCAN (rather than the implementation of DBSCAN) to perform clustering for contact tracing. Since the dataset includes three-dimensional Cartesian coordinates (latitudes and longitudes) not two-dimensional Cartesian coordinates we need to perform trigonometric calculations to obtain Euclidean distances over the earth's curved surface. DBSCAN allows certain parameters which reduce explicit programming and development of clustering code from scratch. The first parameter is epsilon (eps) which assumes the radial distance; which can be set to minimum safe distance dictated by government/administrative bodies to prevent the transmission of the COVID-19 virus. Furthermore, since we have to evaluate three-dimensional Cartesian distances among geographical coordinates, another argument metric (metric) allows setting has $rsine$ function that computes three-dimensional coordinates.

4.1 K-Means result

Before obtaining clusters with the K-means algorithm, we require optimal cluster numbers that would be passed as a parameter to the K-means model. The most common and effective way of obtaining the optimum cluster numbers is via Elbow method which works by evaluating the output of the cost function for arbitrary values of k i.e. number of clusters. We usually plot a graph between the numbers of clusters (k) on $x\text{-}axis$ and cost function values on y-axis. The graph appears to be flattening for a critical value of k; at this point an elbow appears to be forming in the

plotline. From Fig. 8, we can observe graph appears to be flattening at $k = 6$.

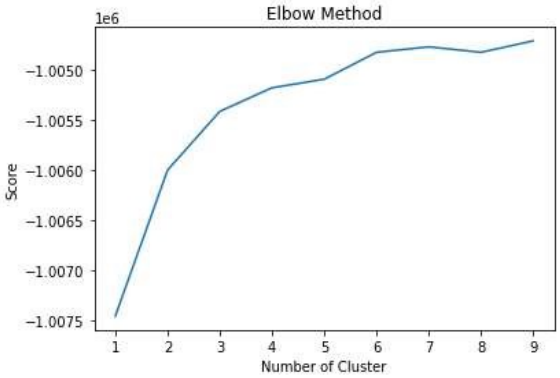

Figure 8. Optimal number of the cluster for K-means generated by Elbow method

Then we can obtain cluster formations from K-means with the number of clusters as 6. In Fig. 10, we can observe 6 clusters. In most of these clusters, there are several data points with distances greater than 6 feet. We can make this remark by comparing cluster data points with the data set locations. This is the major drawback as in our model formulation, we discussed that each formed cluster would be assumed to have samples with a distance of approximately 6 feet among them. Therefore, we can safely assume that to represent the correct relationship between sample points, the number of clusters must be increased. We evaluate the distances among the sample points of each cluster formed during every iteration of K-means while increasing the number of clusters gradually. We arrive at a convincing state when the number of clusters was in the range 25-35, which can be clearly observed from Fig. 9.

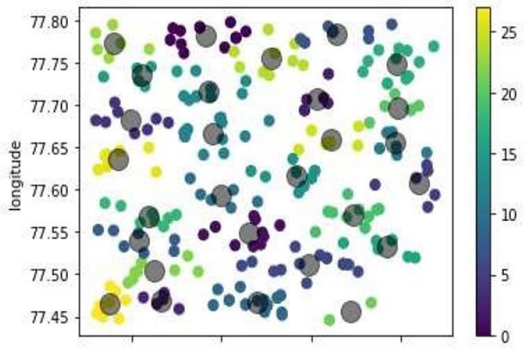

Figure 9. Cluster number=28

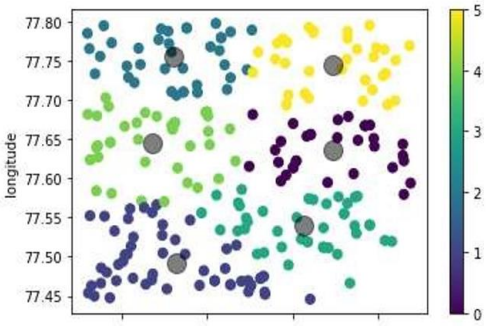

Figure 10. Cluster number=6

4.2 DBSCAN Result:

Similarly, we generated clusters with DBSCAN. It requires parameter *eps* (the similarity between the sample points), *min_samples* (minimum number of sample points to form a relationship), and metric (the types of distance measure). After providing all these parameters with appropriate values, DBSCAN generated 28 clusters, each cluster containing a sample point with the distance of approximate distance of 6 feet between them. Cluster formation is shown below in Fig. 11.

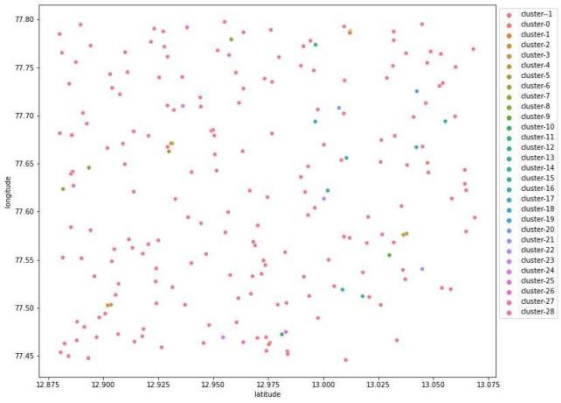

Figure 11. Cluster formation of DBSCAN with eps=6 feet

After obtaining clusters from both clustering approaches, we can execute the rest of the contact tracing approaches discussed earlier. Fig. 12 depicts the infected list with patient 0 as Sarsij.

```
Sarsij can spread Virus to ['Ramesh']
Manish can spread Virus to []
Arminder can spread Virus to ['Adarsh', 'Sristhi']
Vivek can spread Virus to ['Uma']
Lucky can spread Virus to ['Mahesh']
Adarsh can spread Virus to ['Sristhi', 'Arminder']
Sristhi can spread Virus to ['Adarsh', 'Arminder']
Uma can spread Virus to ['Vivek', 'Ramesh']
Mahesh can spread Virus to ['Lucky']
Ramesh can spread Virus to ['Uma', 'Sarsij']
```

Figure 12. List of Infected individuals with patient 0 as Sarsij

4.3 Discussion

After analysing all the aforementioned results and performances we are in a position to discuss the differences, merits, and demerits of DBSCAN over K-Means and Hierarchical clustering methods. Listed below are primary differences among them:

K-Means requires a parameter k i.e. the number of clusters to initiate the clustering process and the same for Agglomerative clustering whereas no such parameter is required for DBSCAN. In the real world, it's an obstacle to obtain optimal cluster numbers to evaluate the most efficient cluster formations. Therefore, for our use case of contact tracing DBSCAN is better suited.

DBSCAN can generate any arbitrary shape of clusters depending on the statistical nature of the dataset distribution whereas K-Means and Agglomerative Clustering can generate only spherical shaped clusters. The real-world movement and thus the location of people can be arbitrary hence DBSCAN performs well over this dataset type and generates cluster formations that accurately reflect the dataset.

DBSCAN cannot cluster datasets with the large difference in densities (sparsely located sample points) because the min_sample parameter cannot be selected. However, since for this work's use case of contact tracing, the minimum transmission distance for contagion spreading is of the order of 6-10 feet, this specific drawback of DBSCAN doesn't raise any concerns as far the distance as lower the chances of virus transmissions.

Hierarchical Clustering breaks down larger clusters into smaller ones, but DBSCAN maintains it into a larger cluster which also doesn't affect our use case in any sense whatsoever.

5. Conclusion and Future Scope

Due to the rise in severity of the COVID-19 pandemic, it is of utmost importance that in addition to existing healthcare provisions and systems, newer methods boosted by recent technologies must be exploited to prevent further loss of life. Artificial Intelligence is one such technology that can help such efforts. In this paper, the effectiveness of certain clustering algorithms in COVID-19 contact tracing is analysed. We can surmise that DBSCAN performs well and is less complex to implement for clustering tasks. In the future with the increment of the computational capacity of scientific as well as consumer device Artificial Intelligence, based operations will be implemented with greater efficiency. Algorithms with the help of enhanced computing machines will be able to identify and extract a greater amount of patterns and thus will be able to categorize and cluster data points with greater precision and logic which will surely help in contact tracing applications.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] R. Vaishya, M. Javaid, I. Khan, and A. Haleem, "Artificial Intelligence (AI) applications for COVID-19 pandemic", Diabetes & Metabolic Syndrome: Clinical Research & Reviews, vol. 14, no. 4, pp. 337-339, 2020. Available: 10.1016/j.dsx.2020.04.012.

[2] M. Kretzschmar, G. Rozhnova, M. Bootsma, M. van Boven, J. van de Wijgert and M. Bonten, "Impact of delays on effectiveness of contact tracing strategies for COVID-19: a modelling study", The Lancet Public Health, vol. 5, no. 8, pp. e452-e459, 2020. Available: 10.1016/s2468-2667(20)30157-2.

[3] P. Rai and S. Singh, "A Survey of Clustering Techniques", International Journal of Computer Applications, vol. 7, no. 12, pp. 1-5, 2010. Available: 10.5120/1326-1808.

[4] Hammouda, Khaled, and F.Karray, "A comparative study of data clustering techniques." University of Waterloo, Ontario, Canada 1, 2000.

[5] J. S. R. Jang, C. T. Sun and E. Mizutani, "Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence [Book Review]," in IEEE Transactions on Automatic Control, vol. 42, no. 10, pp. 1482-1484, Oct. 1997, doi: 10.1109/TAC.1997.633847.

[6] N. Ahmed et al., "A Survey of COVID-19 Contact Tracing Apps," in IEEE Access, vol. 8, pp. 134577-134601, 2020, doi: 10.1109/ACCESS.2020.3010226.

[7] D. Phung, G. Webb and C. Sammut, Encyclopedia of Machine Learning and Data Science. New York, NY: Springer US, 2020.

[8] M. Ankerst, M. Breunig, H. Kriegel and J. Sander, "OPTICS", ACM SIGMOD Record, vol. 28, no. 2, pp. 49-60, 1999. Available: 10.1145/304181.304187.

[9] Sami Ayramo and Tommi Karkainen, "Introduction to portioning-based clustering methods with a robust example", 2006.

[10] T. House and M. Keeling, "The Impact of Contact Tracing in Clustered Populations", PLoS Computational Biology, vol. 6, no. 3, p. e1000721, 2010. Available: 10.1371/journal.pcbi.1000721.

[11] M. Kretzschmar, G. Rozhnova, M. Bootsma, M. van Boven, J. van de Wijgert and M. Bonten, "Impact of delays on effectiveness of contact tracing strategies for COVID-19: a modelling study", The Lancet Public Health, vol. 5, no. 8, pp. e452-e459, 2020. Available: 10.1016/s2468-2667(20)30157-2.

[12] McInnes et al, (2017), hdbscan: Hierarchical density based clustering, Journal of Open Source Software, 2(11), 205, doi:10.21105/joss.00205
[13] L. McInnes, J. Healy and S. Astels, "hdbscan: Hierarchical density based clustering", The Journal of Open Source Software, vol. 2, no. 11, p. 205, 2017. Available: 10.21105/joss.00205.HDBSCAN; Copyright 201, Leland McInnes, John Healy Steve Astels Revision 109797c7.
[14] R. Jain, M. Gupta, S. Taneja and D. Hemanth, "Deep learning based detection and analysis of COVID-19 on chest X-ray images", Applied Intelligence, vol. 51, no. 3, pp. 1690-1700, 2020. Available: 10.1007/s10489-020-01902-1.
[15] V. Omanashvili, "JSON Generator – Tool for generating random data", JSON Generator, 2021. [Online]. Available: https://www.json-generator.com/.
[16] "Building a Simple Contact Tracing Model Using the DBSCAN Algorithm", Medium, 2021. [Online]. Available: https://medium.com/swlh/building-a-simple-contact-tracingmodel-using-the-dbscan-algorithm-5ea796d7afdc.
[17] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD’96), 1996, pp. 226–231.
[18] Sarki R, Ahmed K, Wang H, Zhang Y. Automated detection of mild and multi-class diabetic eye diseases using deep learning. Health Inf Sci Syst. 2020 Oct 8;8(1):32. doi: 10.1007/s13755-020-00125-5. PMID: 33088488; PMCID: PMC7544802.