Nearest Blood & Plasma Donor Finding: A Machine Learning Approach

Nayan Das, MD. Asif Iqbal
Department of Computer Science and Engineering (CSE)
Chittagong University of Engineering and Technology (CUET), Chittagong, Bangladesh
nayan271997@gmail.com, asifiqbalsagor123@gmail.com

***Abstract*—The necessity of blood has become a significant concern in the present context all over the world. Due to a shortage of blood, people couldn't save themselves or their friends and family members. A bag of blood can save a precious life. Statistics show that a tremendous amount of blood is needed yearly because of major operations, road accidents, blood disorders, including Anemia, Hemophilia, and acute viral infections like Dengue, etc. Approximately 85 million people require single or multiple blood transfusions for treatment. Voluntary blood donors per 1,000 population of some countries are quite promising, such as Switzerland (113/1,000), Japan (70/1,000), while others have an unsatisfying result like India has 4/1,000, and Bangladesh has 5/1000. Recently a life-threatening virus, COVID-19, spreading throughout the globe, which is more vulnerable for older people and those with pre-existing medical conditions. For them, plasma is needed to recover their illness. Our Purpose is to build a platform with clustering algorithms which will jointly help to provide the quickest solution to find blood or plasma donor. Closest blood or plasma donors of the same group in a particular area can be explored within less time and more efficiently.**

***Keywords*—Blood donation, Plasma donation, K-means clustering, Labeled Agglomerative clustering**

I. Introduction

Blood is a liquid substance that circulates the necessary nutrients and oxygen to every cell of the body. Every year many people died because of the shortage of safe blood. This problem becomes severe during pregnancy and major operation. Again blood requires for some other diseases like Cancer, Dengue and Leukemia, thalassemia, etc. Over a million people died because of dengue in Bangladesh. There are time periods in which blood must be processed or preserved. Otherwise, collected or stored blood can't be used. So, An efficient flow of blood is required in blood bank or blood donation camps to meet the regular demands of recipients. Recently concern grows about the plasma donation for COVID-19 during the pandemic situation. This convalescent plasma was used to recover patients who are critically ill as it helps to grow antibodies on their body [1].

People all over the world donate blood on these purpose. According to the World Health Organization (WHO), around 118.5 million blood donations are collected globally, 40% of which are collected from high-income countries [2]. Many organizations help blood donors donate blood and plasma via many applications and online social groups. But this application and online social groups remain analog, and we need the quickest solution in this regard. In regular blood donation applications and social groups, people share their needs for blood and get some information lately that can be less useful in an emergency condition.

The key objectives of our work are-

- To build a platform between blood donor and receiver.
- To implement a hybrid approach of K-Means and Agglomerative clustering algorithm.
- To find the nearest blood donor in a specific region in the shortest possible time.
- To increase the number of voluntary unpaid blood donations significantly.

Different methods have been used to solve this problem. This time, we have tried another way, a clustering approach, to solve the problem by grouping every user into small groups. This unsupervised machine learning approach is much faster and effective.

In section II, we will discuss related work done previously to solve this problem. In section III, clustering algorithms relating to our project will explicitly be discussed. In section IV, our proposed method will be presented. In section V, we will analyze our experiment result.

II. Related Work

To find the solution of the donor finding problem, we have analyzed some previous work regarding this context. We got some ideas from all these works and tried to figure out the advantages and shortcomings. We tried to overcome all those difficulties. Different methods and platforms have been used to solve this problem. T. Alanzi et al. [3] have used social media as a platform to sort the process of blood donation. Social media may solve the problem, but it can't find the nearest blood donor and may not meet the specific seeker's requirements. This approach can also be more time-consuming. T. Wangchuk et al. [4] have made a survey analysis through some online questionnaires. They analyzed that separate mediums like social platforms couldn't meet the demand quickly, whereas a mobile application could. V.K. Tatikonda et al. [5] have made a web application to find blood donors from which an authorized clinic can make a request. Unfortunately, this application also couldn't find the nearest blood donor. M.S. Sabir et al. [6] have proposed a method to find the most immediate blood donor using the Dijkstra algorithm, which is considerably less efficient for a vast number of user datasets and also requires a tremendous amount of execution time. Finally, H.D. Das et al. [7] have developed a system using GPS to find blood donors of a limited area. They have worked on a limited number of datasets, which may be more time consuming with the increase of datasets. We have tried to apply a machine learning approach to make it quicker and more user-friendly to solve all this problem.

III. Background study

Our approach is based on clustering, which involves grouping a collection of objects such that objects are grouped in different clusters. Clustering is widely used in many fields in the real world. There are many clustering algorithms. We will use two of these algorithms, K-means clustering, and Agglomerative clustering. A short discussion is given below about those algorithms.

A. K-means Clustering

K-means clustering is the most straightforward clustering algorithm in unsupervised machine learning algorithms. K-means clustering clustered iteratively. At first, it selects some random centroids [8]. These centroids are the beginning point of k-means clustering. It then clustered the closer points into the centroids and made them clustered. Figure 1 depicts the initial centroid being selected.

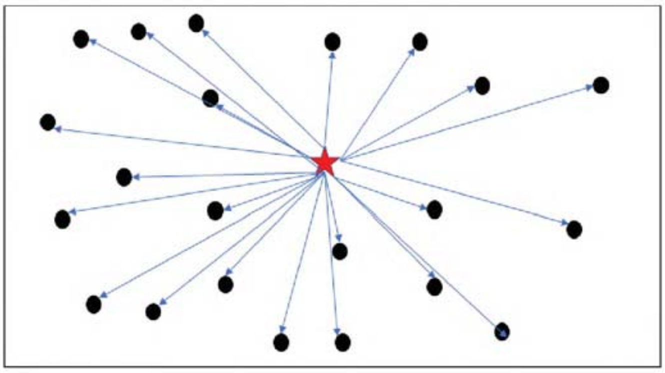

Figure 1. Centroid being selected

It iteratively shifts the centroids to get the better cluster. Figure 2, figure 3 depict the process of centroid shifting. It halts after

(a) the centroid is fixed- No change or a little in the centroids any more
(b) completion of a fixed iteration

Figure 4 depicts the final cluster being made.

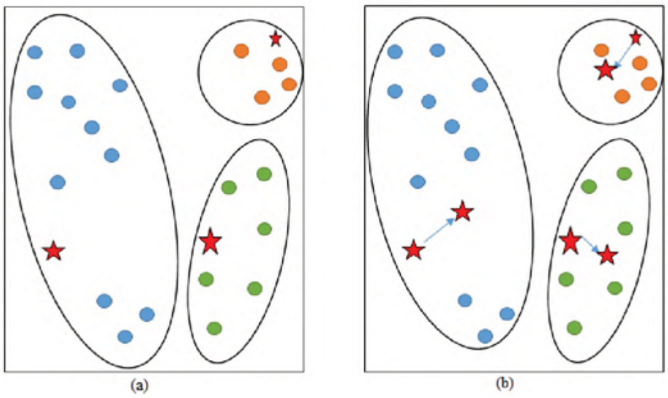

Figure 2. a) Initial Centroids b) Shifting Centroids

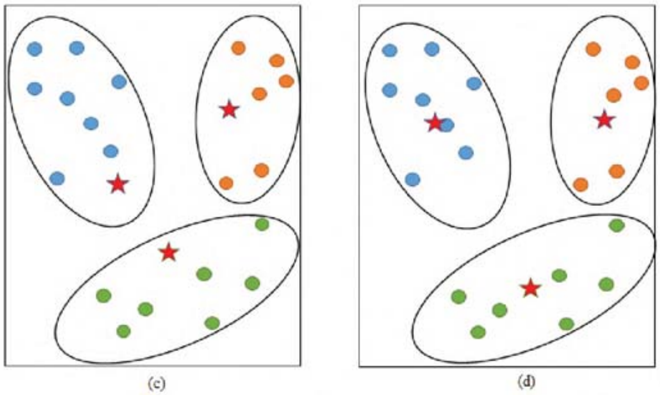

Figure 3. c) Shifted Centroids d) New Centroids

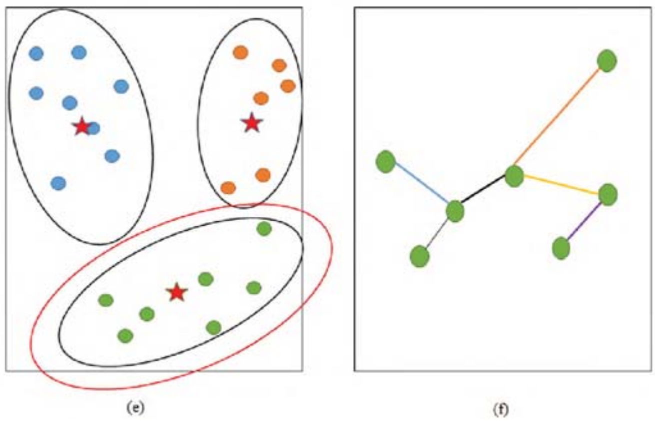

Figure 4. e) Final Cluster f) Relation Between Cluster node

This clustering is much appreciable for a large data set. The standard k-means algorithm's time complexity is $O(nkt)$, where n is the input data size, k is the number of clusters, and t is the algorithm's iteration. But S. Na et al. improved the complexity of k-means to $O(nk)$, where n is the input data size, and k is the number of clusters [9].

B. Agglomerative clustering

Agglomerative clustering is a hierarchical clustering that builds a cluster by using a bottom-up approach [10]. This algorithm makes clusters by making a singleton cluster and ends up making a single cluster. A data is included in the singleton cluster based on Euclidean distance. The resultant tree-based representation is known as Dendrogram.

Agglomerative clustering's time complexity is $O(n^3)$, where n is the input data size [11]; thus, this algorithm is very much time-consuming. So, if the input data size increases, it will consume more time to cluster. We use labeled Agglomerative clustering. Here cluster is labeled according to a fixed desired donor.

For example, in figure 5, we labeled the agglomerative cluster with five donors. So, the five closest donors will be label 0. Next, five donors will be label 1; the next five donors will be label two and go on. Thus we can separate the closest donors from the distant donor, according to our needs.

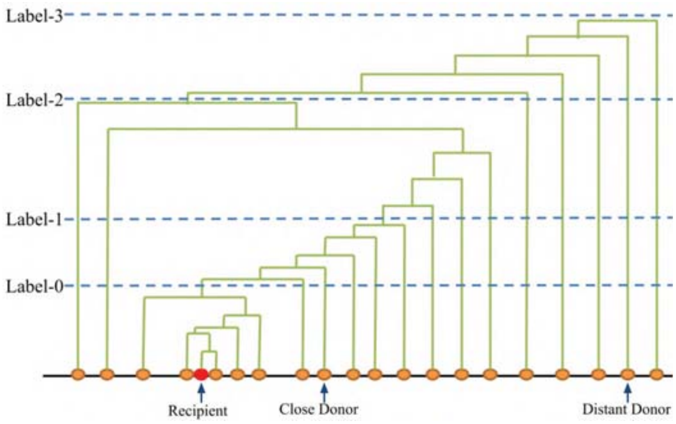

Figure 5. labeled Agglomerative Cluster

IV. Methodology

Our whole project is based on finding blood or plasma donors more efficiently. In Figure 6, a flow of the entire work is shown. For simplicity, the whole process will discuss into step by step process.

A. Making Initial Cluster

There are two types of clustering process used in this project. The first one is introduced as an initial cluster. This cluster is made by using the k-means algorithm. In an urban area, there are many blood donors in different locations. Some of the blood or plasma donors share a particular region where they stay. So the first approach is to make those groups of people separated by an initial cluster by their location. Every donor location is converted to its corresponding latitude and longitude form.

Latitude and longitude are the geographical coordinate system used worldwide to determine every part of the world through a specific set of numbers. For converting location into a geographical coordinate, we use API from OpenCage Geocoder [12]. It is a free, open, and most reliable API for conversion to geographical coordinates. The geographical coordinate is appended on the database when the user will register as a donor. We use latitude and longitude of donors to make the initial clusters. The number of clusters being calculated automatically by the number of donors admin wants on a particular region. We select the number of the initial cluster by the following equation

$$C = \frac{D}{DR}$$

Where, C = Number of Cluster

D = Number of Total Donor

DR = Number of Donor admin wants on a particular region

A donor who gave blood within three months is temporarily discarded from the database so that the capable donor may get preference. In the case of plasma, as the donor can donate plasma every 21 days, after donating plasma donor will be disabled for 21 days [1]. This cluster is made with a predefined delay of time. This delay can be a few days, which selects by the admin manually. As we make the initial cluster with latitude and longitude of donors, the centroids of the initial k-means are also related to the latitude and longitude—the centroids of the initial k-means cluster stored for the next steps.

B. User Request

The recipient can request a donor. S/he can ask anytime through our system. Every recipient must need to be registered to the system before making any request. The recipient needs to give information about his/her blood group and necessary information like location and cellphone number by registering. The registered recipient can be selected as a donor for future requests. If the recipient is recorded once, she/he can request the central server through the application. For searching a donor, The recipient needs to share his desired location where blood or plasma is required e.g., "X hospital" or "Y clinic," and wait for the response report. In response, the server will send him/her specific donor information.

C. Making Final Cluster By Agglomerative Clustering

This is called a final cluster, as it is the efficient cluster used for searching blood or plasma donors. For making this last cluster, we use agglomerative hierarchical clustering. According to the blood or plasma location needed, we extract the geolocation of the given area by the recipient. The geolocation is in latitude and longitude form, which represents the location of the recipient. According to the recipient's latitude and longitude, distance from the centroids of the initial k-means clusters is calculated.

For comparing the distance, geodesic being used. Geodesic is a length-minimizing curve which are circles in the sphere but straight lines on the plane. Python has a dedicated package for geodesic.

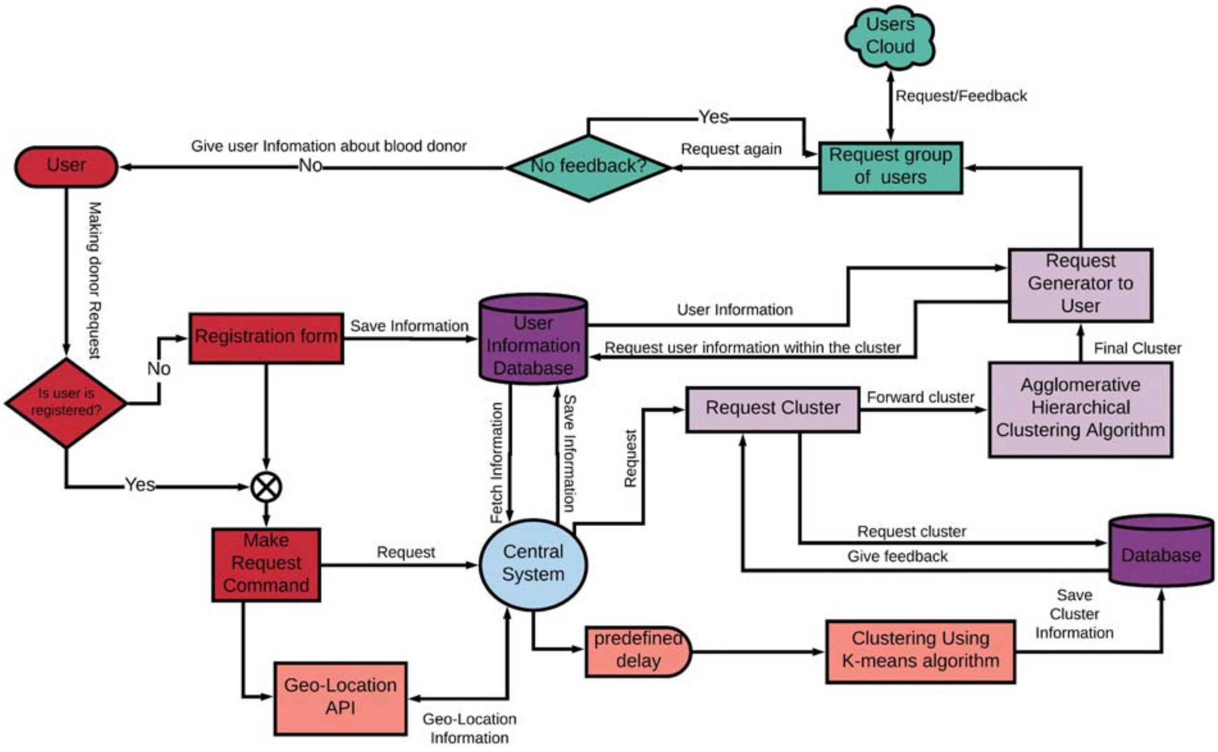

Figure 6. Proposed Methodology

After comparing the closest centroids of the initial k-means centroids with the recipient geographical coordinate, the final cluster selected. Information regarding selected cluster separated from the main database. Every donor's distance in the chosen cluster is calculated by using geodesic and merge with the selected cluster.

Using the geographical coordinate with distance from the selected cluster's recipient, a labeled hierarchical cluster is made by agglomerative clustering. As Label increases, the donor will be far from the receiver. Figure 5 depict the scenario clearly. The low labeled donors will get high priority as they are closer than any other label.

D. Request Donor

The selected cluster dataset is now divided by the labels made by agglomerative clustering. Every label is categorized with label-0 to label-n with comparing the distance from the recipient to the donor. Here n is the number of the label made by the agglomerative cluster.

We need to find a specific donor by requesting every donor within the label. The final labeled donor's information fetched and then asked every donor in the labeled donor list for donation. The request is made by making a notification on their devices. If they didn't back any response or were not interested in giving blood or plasma, we will go to the next label donor list. If they are interested, then we will forward their information to the recipient as a donor.

A sequence diagram of our proposed method is given in Figure 7. It depicts a scenario of how a server responds with a donor request with its pre-request and post-request process to solve the problem. Half of the processing is predefined and the rest of the processing is done after the recipient request.

V. Experimental Result & Analysis

After cleaning and pre-processing a blood donor dataset with 2375 records was used for experimental results and analysis. After cleaning, Eight hundred seventy-five records of this dataset were contributed by a local blood donor camp from Chittagong, Bangladesh. The rest of the 1500 records were provided by another blood donor camp from Dhaka, Bangladesh. Name of the Donor, Mobile Number, and other sensitive information was discarded from the dataset by authorities. In figure 8, an open street map is shown with user latitude and longitude to show user distribution across Bangladesh. This open street map is drawn on the tableau desktop version. Usually, in the urban area, there are many donors than in rural areas.

Experiments were run on a computer as server with an Intel® Core$^{\text{TM}}$ i5-4200H processor running at 2.80 GHz using 8 GB of RAM, running on Windows 10 Pro version. GPU specification is NVIDIA GeForce GTX 950M with 4GB RAM.

A. Experimental Result

The proposed method has two clustering algorithms used to find donor information named k-means and ag-

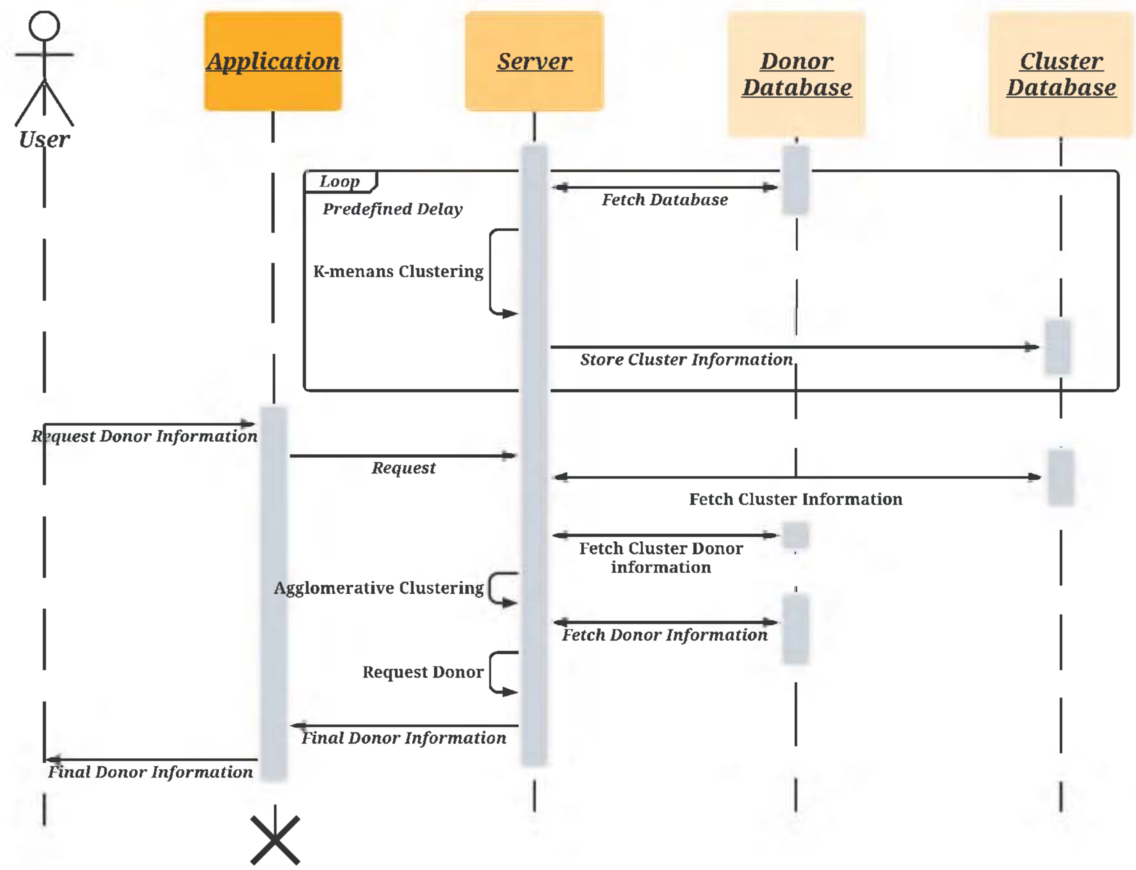

Figure 7. Sequence Diagram of Proposed Method

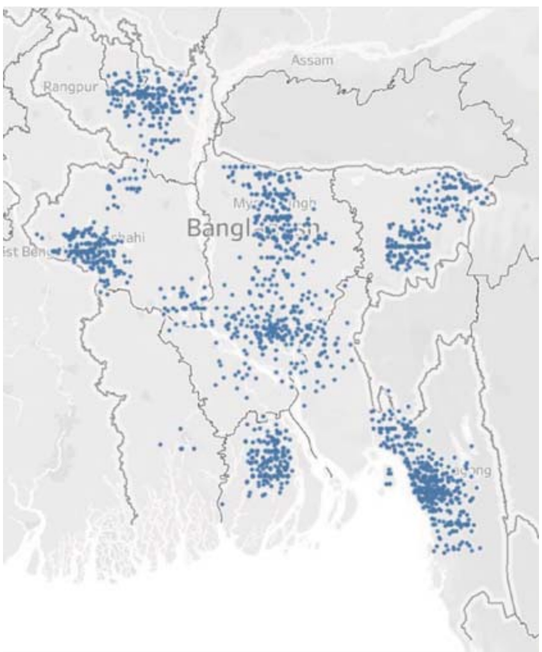

Figure 8. Blood Donor's Dataset Location Distribution across Bangladesh

glomerative clustering.

As depicted in figure 7, cluster through k-means is done previously, but the agglomerative clustering needs to do with every search of the recipient. In this subsection, a brief description of the required time to search each donor will discuss.

- **K-means clustering time:** For making the initial cluster through k-means required time. K-means cluster required less time as it's complexity is $O(nkt)$.
- **Agglomerative Clustering time:** Every request by the recipient needs to perform agglomerative clustering. By doing labeled agglomerative clustering, we can prioritize the closer donor for donating blood.
- **Requesting the Donor:** A request to every closest donor for donating blood is made by the server to finalize the final donor. This process needs an arbitrary amount of time.

We implement our proposed method with our cleaned dataset of Blood Donor. The time for K-means clustering is given in table I. On the table, We can see that the required time is gradually rising by increasing the number of clusters.

On every donor request, Agglomerative clustering is done by the server to find the closest available donor. For

Table I
K-MEANS CLUSTERING TIME (IN SECONDS)

	Number of Cluster					
	4	5	6	7	8	9
Time	0.0884	0.1092	0.1105	0.1239	0.1287	0.1637
	Number of Cluster					
	10	15	20	25	30	35
Time	0.1822	0.2334	0.2886	0.3466	0.3880	0.4260

different numbers of k-means clustering, agglomerative clustering takes different time. By assuming the recipient's location as "Chittagong Medical College, Chittagong, Bangladesh ", the execution time for each agglomerative clustering for different k-means cluster is shown in the table II.

Table II
AGGLOMERATIVE CLUSTERING TIME (IN SECONDS) CONCERNING K-MEANS

K-means with 5 cluster

	Agglomerative Clustering label					
	7	8	9	10	11	12
Time	0.1239	0.1353	0.1262	0.1272	0.1265	0.1274

K-means with 10 cluster

	Agglomerative Clustering label					
	7	8	9	10	11	12
Time	0.0977	0.0983	0.0990	0.0971	0.1001	0.1047

K-means with 15 cluster

	Agglomerative Clustering label					
	7	8	9	10	11	12
Time	0.0724	0.0724	0.0723	0.0642	0.0722	0.0813

K-means with 20 cluster

	Agglomerative Clustering label					
	7	8	9	10	11	12
Time	0.0306	0.0476	0.0314	0.0326	0.0342	0.0354

K-means with 25 cluster

	Agglomerative Clustering label					
	7	8	9	10	11	12
Time	0.0237	0.0238	0.0239	0.0242	0.0245	0.0285

B. Analysis

After the experiment, we analyzed our proposed method to find a more efficient way of getting the closest donor. As, table II shows, for a query from the recipient, execution time needs less than 135 ms maximum. In figure 9, a time comparison between k-means clustering and agglomerative clustering is shown. On that figure 9, there is a variable number of clusters of k-means ranging from 0 to 35, and the agglomerative cluster has 5 clusters. As the Number of k-means cluster increases, agglomerative cluster taking less time to find the closest blood donor.

VI. CONCLUSION

In our whole project, we have built a platform for the blood donor and receiver. We have combined two well-

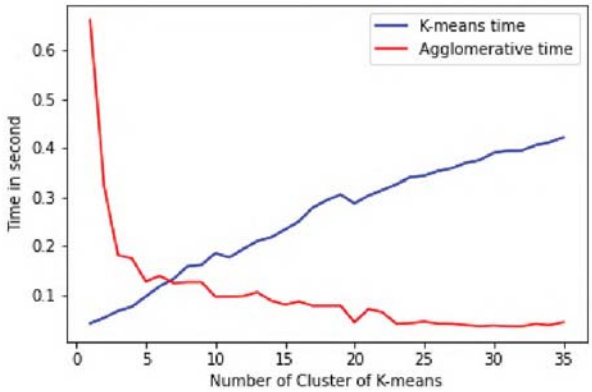

Figure 9. Time comparison between variable K-means and fixed Agglomerative

known clustering algorithm, k-means, and agglomerative clustering, to solve the problem. From experimental results analysis, we can descry that our system gives a more robust and efficient result in this regard. Our sole purpose is to reduce the time as well as the sufferings of the people. A blood or plasma recipient can efficiently get donors using our system in any situation.

REFERENCES

[1] X. Chen, "Commercial plasma donation and individual health in impoverished rural china," *Health economics review*, vol. 4, no. 1, p. 30, 2014.

[2] "Blood safety and availability," https://www.who.int/news-room/fact-sheets/detail/blood-safety-and-availability, (Accessed on 09/05/2020).

[3] T. Alanzi and B. Alsaeed, "Use of social media in the blood donation process in saudi arabia," *Journal of Blood Medicine*, vol. 10, p. 417, 2019.

[4] T. Wangchuk, K. Wangmo, U. Wangchuk, P. Gyem, P. R. Dhungyel *et al.*, "Need of medium for finding blood donor in bhutan," *Asian Journal For Convergence In Technology (AJCT)*, 2018.

[5] V. K. Tatikonda and H. El-Ocla, "Bloodr: blood donor and requester mobile application," *Mhealth*, vol. 3, 2017.

[6] M. S. Hossain, N. Das, M. K. H. Patwary, and M. Al-Hasan, "Finding the nearest blood donors using dijkstra algorithm," *SISFORMA: Journal of Information Systems (e-Journal)*, vol. 5, no. 2, pp. 40–44, 2019.

[7] H. D. Das, R. Ahmed, N. Smrity, and L. Islam, "Bdonor: A geo-localised blood donor management system using mobile crowdsourcing," in *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, 2020, pp. 313–317.

[8] S.-Q. Wang and D.-M. Zhu, "Research on selecting initial points for k-means clustering," in *2008 International Conference on Machine Learning and Cybernetics*, vol. 5. IEEE, 2008, pp. 2673–2677.

[9] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *2010 Third International Symposium on intelligent information technology and security informatics*. IEEE, 2010, pp. 63–67.

[10] K. Sasirekha and P. Baby, "Agglomerative hierarchical clustering algorithm-a," *International Journal of Scientific and Research Publications*, vol. 83, p. 83, 2013.

[11] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[12] "Opencage geocoder." [Online]. Available: https://opencagedata.com/