Predicting COVID-19 Infection Groups using Social Networks and Machine Learning Algorithms

Kyle Spurlock
School of Engineering and Computer Science
Morehead State University
Morehead, KY, USA
kdspurlock@moreheadstate.edu

Heba Elgazzar
School of Engineering and Computer Science
Morehead State University
Morehead, KY, USA
h.elgazzar@moreheadstate.edu

***Abstract*— Today, social media has grown in usage to the point where it is often deeply intertwined with life offline. People share their thoughts, passions, and lives online, and in many ways, these social networks can be considered abstractions of real-world society. The idea for this research is that by modeling on these social networks, these glimpses into people's lives through their words and posts are capable of showing their current health situation, and their susceptibility to outside influences affecting it. The goal of this research project is to design and implement unsupervised machine learning techniques to group together sub-networks of connected individuals in hopes that it may be beneficial to current disease surveillance systems. Using Python programming language and the tools available to it, data was collected from the social network platform Twitter and analyzed using three clustering and centrality measurements. The criterion to be included in the data found tweets containing symptomatic keywords, like those of which experienced by people afflicted with the novel coronavirus disease (COVID-19). It is our findings in this research that by simulating the real-world connections that people have with their surrounding cliques using the ones that they exist within the virtual world, new possibilities for viral control and disease prevention become available using easily sourced, and quickly gatherable information.**

Keywords*— *COVID-19, clustering, unsupervised machine learning, Twitter, disease surveillance, social networks, and centrality

I. Introduction

The usage of social media in many of the countries of the world has grown significantly over the recent decade. These underlying networks of people sharing and posting their thoughts, feelings and lives and can almost be considered underlying societies in and of themselves. What is more beneficial is that a large part of this shared information is easily extractable from these networks and can be further used to determine subgroups of people that exist within these virtual worlds. This collected information is commonly used for commercial purposes such as ad suggestions, but this research aims to look at an alternative use for this data in viral disease prevention.

At the time this research work was conducted, the world was facing a wide-spread pandemic from the SARS-CoV-2 (COVID-19 or novel coronavirus) virus. Believed to originate in Wuhan, China, the virus was able to make its way around the globe in a matter of months, forcing many countries to put in place restrictive measures to reduce the spread of the virus. Throughout the 20th and 21st century, there have been many outbreaks of infectious disease that have devastated parts of the world, through such conditions as severe acute respiratory syndrome (SARS), H1N1 influenza, and Ebola. To combat these outbreaks, often surveillance systems have been employed to record information relating to a virus that can then be analyzed to aid in ongoing and future prevention. One such surveillance system is the National Notifiable Disease Surveillance System (NNDSS), operated by the Center for Disease Control (CDC) [2]. While this large-scale system is responsible for the collection and distribution of information regarding various notifiable and reportable diseases, it ultimately has shortcomings that can impede the effectiveness of its purpose [2]. National-level reporting systems such as these often depend on the state and local levels to provide accurate data, in order to collate it into a nation-wide collection. In the case of the NNDSS it is optional for these states to share information with the CDC, which may result in missing or incomplete records of an outbreak [2]. Some additional challenges that come with implementing a health surveillance system depends on the country that seeks to implement them. Some low- and middle- income countries simply do not possess the health infrastructure to execute surveillance across the country using conventional means [3]. Other challenges may come from delayed and skewed reporting, as a test result may take a considerable amount of time before it becomes reportable, and patients with milder symptoms may seek medical care less often, skewing how widespread the disease is [1].

Ideally, it is the intention of using such universally available information to assist, but not wholly replace, contemporary health surveillance techniques. Either this research and its methodology could be used as a model for applying similar techniques to collected patient health databases, or it could be used in tandem with premonition acquired from medical data to discover high risk groups around a patient zero. Compared to building additional infrastructure to accommodate wide ranging surveillance programs, it is far more cost effective to utilize readily available information that the subjects of surveillance can provide directly. It is a simple process to request access to the API of many of the most popular social networking sites, such as Twitter through their Twitter Developer program. This can allow for an individual alone to collect tens of thousands of users' information with minimal downtime, that can then be further analyzed to produce more beneficial statistics and figures.

In this work, we have examined a new approach to utilizing unsupervised machine learning techniques to supplement and advance big data surveillance for the benefit

of public health. The data used in this paper has been collected from scratch through Twitter and has been clustered using spectral, agglomerative hierarchical, and K-means clustering. Key figure analysis using multiple centrality measures has also been performed to detect notable members of communities.

The following sections will look at how information can be collected from these social sites, and how this data is able to be turned into processible datasets that can then be utilized with clustering and centrality assessments to discover cliques and the important users that exist within them. First, homage will be paid to the researchers that have previously worked on similar topics, and differentiations will be discussed from this research to demonstrate how these ideas have advanced into our current topic. Afterwards, an overview of the methodology, our results, as well as plans for the future usage of these methods will be discussed.

II. Related Work

There exist multiple studies examining the potential of big data for preventative health measures, and these studies have been conducted by researchers from all around the world. These types of studies often will consider search trends and queries to detect the presence of potential viral outbreaks occurring within a community. As for how this differentiates with this research, rather than looking at an abstract view of where a virus may be spreading, the data utilized in this paper has been collecting in a similar way that a smart advertising algorithm may work. This allows for users of interest to be identified directly along with their surrounding social groups.

One such study conducted by Chae, Kwon, and Lee, sought to examine four different categories of data and how they correlated to the presence of malaria, chickenpox, and scarlet fever within South Korea [4]. Among these categories was big data collected from Twitter, and internet search queries collected from the Naver Data Lab, which is the most popular search engine in South Korea [4]. This data in combination with actual infectious disease figures was then utilized with deep learning models to predict outbreaks in a timelier manner than that which could be reported by the Korea Center for Disease Control (KCDC) [4].

A study within a similar vein was conducted by Shin et al. sought to do similar work by finding the connection between Twitter and Google trends within Korea during the Middle Eastern respiratory syndrome Coronavirus (MERS-CoV) outbreak of 2015 [5]. Similar to the previously mentioned study by Chae Kwon, and Lee, this research also wanted to analyze the potential that big data and search trends could bring to digital surveillance. In this study, search terms and tweets relating to MERS and its symptoms were cross referenced with the official release numbers of confirmed cases, and together all three results showed significant correlation to one another [5].

Another study performed in 2015 by Dion, AbdelMalik, and Mawudeku, wanted to provide evidence of the effectiveness big data has on disease surveillance by examining its usage within The Global Public Health Intelligence Network (GPHIN) [25]. The technology and methods to stream a large volume of data through an automated system are utilized by the GPHIN to detect potential warning signs of disease outbreaks through news outlets from around the world [25]. Through using this system, the GHPIN is credited with detecting the first case of MERS-CoV within the middle east, identifying the presence of SARS within China in 2003, and monitoring the Ebola epidemic within West Africa [25].

These studies show that trends appearing in posts, search queries, and even newspaper articles have been shown to be connected to occurrences of disease within an area. However, the predicative ability of knowing where a virus is likely to occur within a group; and who may be most capable of spreading it within this group, is something that has not been fully explored yet. This is where our research seeks to differentiate itself, by determining on a local scale which individuals may be at risk of infection behind the trends.

III. Proposed Methods

The process by which this research was conducted is demonstrated in Fig. 1. Users were collected from Twitter in March-April of 2020, at the beginning of the development of new cases of COVID-19 within western countries.

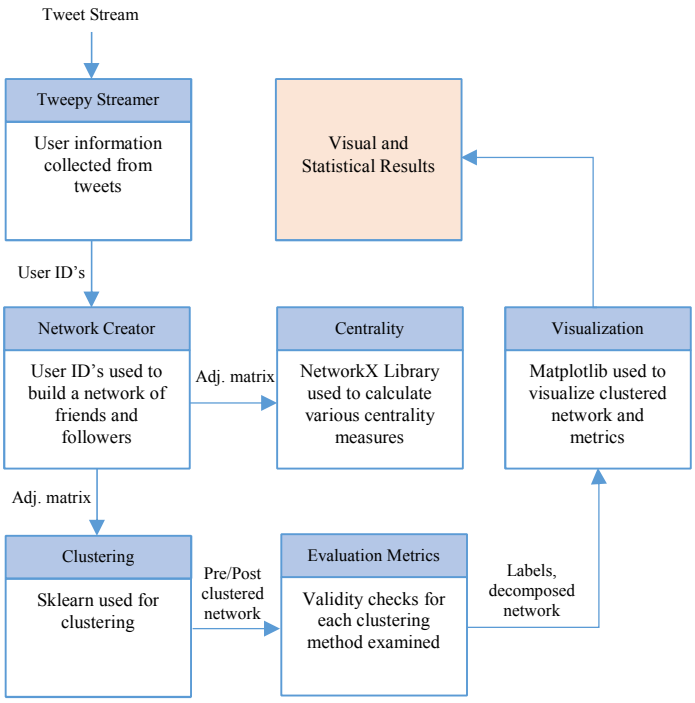

Fig. 1. Process of Proposed Methods Chart

To analyze this data, two unsupervised learning techniques known as clustering and centrality were utilized to find groups of users and the key figures within them respectively. Unsupervised learning refers to the method of analyzing information in a way that considers no acknowledged pre-emptive classifications that may determine the outcome. Clustering is commonly known as the process of finding hidden figures within the data, and after execution, there is no external input that may influence the results. If a specified grouping of users was available before collection, supervised learning techniques could have instead been applied to this trained data to produce a conclusion with even greater efficacy. For this research we have built our algorithms with Python, with various libraries being utilized for different

purposes. Tweepy [6] was used to interface with the Twitter API for data crawling. Matplotlib [8] was used for visualizing cluster count metrics and the clustering graphs. Sci-Kit: Learn [9] was utilized for clustering and decomposition. NetworkX [7] was used for visualization and for centrality measurements. Additionally, the SciPy library [22] was used for specific graphing tools, and the Pandas library [23] for data frame functionality.

A. Data Collection

While there exists a large amount of data sets catered to the analysis of users and their positions within social networks, most public user data sets for social media are tailored largely on the individual features of each user, and focus on features such as their friends and followers counts. This is undeniably useful for selecting the most popular users from a network, but to find the full effect of the network that binds these users together and has the potential to connect their current health situation at the time of the current pandemic, data collection was started from scratch with individual user connections prioritized.

Data was collected from Twitter using the Tweepy [6] library and then converted into an adjacency matrix to be later assessed with clustering and centrality measures. The avenue of collection was through streaming newly created posts while looking for specific keywords such as "Covid-19", "SARS", "coughing," "fever," and similar phrases or hashtags that might suggest a symptomatic individual. The intention behind the selection of keywords was that users may happen to reveal their current health situation based on what they decided to post about, so users that may be feeling sick might make some remark exclaiming that they might have gotten sick with the virus. To further alienate these users of interest, follower and friend range exclusions were introduced to prevent news outlets reporting on the disease from being included, as well as bot accounts that may have an unproportionate following to followed by ratio.

After initial collection stages, a set of 1,000 users that were within the previously described metrics were used to build the network through the friends and followers that they each possessed. An emphasis was placed on collecting the friends/followers that were shared by multiple users, but additional users that would still not be classified as potential media or bot accounts were accepted into the final dataset since they were connected to minimally one user. This created a final network that contained 10,294 nodes with 20,590 edges between them. Additionally, these edges were weighted on a scale of 1 to 3, with 3 representing closest proximity and 1 representing farthest proximity.

Although it would have been feasible to select pre-grouped users based on their individual connections, the random elements associated with streaming the tweets and picking out users of interest means that no initial ground truth or labels were established within this particular dataset, hence the use of unsupervised learning techniques to extract information. If a ground truth could have been established around a particular group known to be connected, geographically or otherwise, supervised learning would have been the preferred option for analysis.

B. Clustering

Clustering forms one of the primary pillars of unsupervised learning and allows for data with unknown correlation to be quantitatively divided into clusters (categories, subsets, or groups), making possible the further study of the connections within these groupings [11]. The definition of the compositional elements of clustering have not been conclusively decided upon, however a general description finds that [10]:

a) Elements within a same grouping must have some measure of homogeneity.

b) Elements in different groupings must be found to be different from one another.

c) The measurement associated with similarity and dissimilarity must be reasonable and have correlation with the labeling of elements.

There exists a large assortment of different clustering algorithms, all associated with these beforementioned principles with differing ways of finding similar and dissimilar elements. For this research, we have chosen to look at spectral, agglomerative, and K-means clustering algorithms, all accessed via the Sci-Kit: Learn library [9]. This selection was chosen out of many possible options due mostly in part to their ease of use, with the bonus that they all possess some metric or visual that allows for the number of clusters to be more easily deduced. Thus, supporting the validity of the chosen cluster counts for each method.

2) Spectral Clustering

Given that the goal of clustering is to identify groups within a network or graph, spectral clustering is an algorithm that utilizes the graph Laplacian to find these groups with high intra-cluster and low inter-cluster similarity [13]. The unnormalized Laplacian of a graph is defined as being $L = D - W$, where D is the degree matrix with degrees along its diagonal, and W is the weighed adjacency matrix of the graph. [12] From the Laplacian, the first k-eigenvectors (k for the number of clusters in the dataset) can be computed using an eigen solver algorithm, this can then be used to find the clustering labels using K-means [12]. Additionally, by graphing the eigen values, the number k of clusters can be deduced by looking at how many eigen values exist before the spectral gap [12].

3) Agglomerative Hierarchical Clustering

Hierarchical clustering is a clustering algorithm that functions exactly as its namesake describes, in that it divides a dataset into a spanning series of hierarchy's with children nodes connected to parent nodes [14]. A dendrogram is commonly used to represent this tree of clusters and how they split apart into larger clusters. The focus of this paper is on agglomerative hierarchical clustering, which entails recursively merging single point clusters with appropriate association until each node in the graph is within one of these clusters [14]. To customize its functionality, a number of different linkage methods such as single, complete, and ward allow for clustering through different metrics [15]. The linkage method used in this research is complete, which measures the maximum dissimilarity between clusters as the means of merger [15].

4) K-means Clustering

K-means is arguably the most popular centroid-based clustering method not only for its ease of use, but for its effectiveness and speed. K-means is referred to as centroid-based because it first identifies k number of cluster centers or centroids, and then assigns nodes in the dataset to clusters

based on their proximity to one of these centers [16]. The process is as follows [16]:

a) Randomly select k entities from dataset to serve as cluster centers.

b) Assign each item in the dataset to a cluster based on the closest cluster center.

c) Calculate the mean of each cluster

d) Repeat until minimal intra-cluster variance is found.

In this research, an iteration of this algorithm known as K-means++ has been used since it selects cluster centers with an improved method outside of random chance. K-means++ works in a similar process to the traditional K-means, however instead of indiscriminately selecting clusters until convergence, a single center is first selected, and then for each other point *x* within the graph the distance from this center is computed, denoted as *D(x)* [17]. *D(x)* is then used in a probability function to determine the next selected center [17]. This process is repeated until *k* clusters have been selected, and then the traditional K-means algorithm is instantiated. [17]

A visualization of K-means that is commonly used to uncover the number of clusters using the method involves graphing the within cluster sum of squares (WCSS) against the number k of clusters[24].

C. Centrality Measurements

Centrality is another one of the fundamental foundations of unsupervised machine learning. While clustering deals with the grouping of similar users, centrality is used to find the central or key figures that exist within these groups [18]. Like clustering algorithms, there exists a plethora of different methods that locate these key figures using different metrics. The methods we have selected for examination in this research are closeness, degree, and page rank centralities. The hopes with determining these central figures in the context of this dataset is that users who have the highest centrality scores may be the prime individuals to surveil, as they potentially might become super spreaders within their associated collectives.

1) Closeness Centrality

Closeness centrality is a method of key figure selection that considers the inverse of the average distance one node has with all other nodes in the network [19]. Its namesake describes exactly what metric it measures, in that the most important nodes are described as having the most nodes within their immediate vicinity [19]. The usefulness of this algorithm in this context can be seen in detecting the users with the highest amount of neighboring users, independent from immediate connections. This would be capable of representing individuals that if infectious, would be most likely to transmit disease to the group they reside in, as well as the groups surrounding them.

2) Degree Centrality

Considered the simplest of the centrality measurements, degree centrality operates on a simple philosophy that the most important nodes must be the ones that have the most connections within a graph [18]. To showcase its simplicity, equation (1) represents this measurement mathematically. Where *G* is a graph with vertices in the graph represented *V(G),* with the degree centrality measurement for any $v \in V(G)$ given by [18]*:*

$$C_D(v) = \frac{d(v)}{|V(G)| - 1} \quad (1)$$

Where *d(v)* is the degree of a given vertex, and $|V(G)|$ represents the total number of vertices within the graph *G*. This method has usefulness in determining the most locally connected users, and how disease may spread through connections from these users.

3) PageRank Centrality

Originally an algorithm designed for ranking web pages by Sergey Brin and Larry Page in 1998, PageRank selects important nodes in a network by considering their immediate connections to other nodes, as well as the connections that those child nodes have to others [20]. It utilized this concept in web page ranking by treating individual website pages as nodes, and the links that connected them to other sites as the vertices [20]. In a way it can be seen as an advancement on the degree and eigen centrality formulas, as not only are the initial nodes connections counted, but the connections that branch from the first set. To differentiate itself from eigen centrality, it also takes into account the weight of connections and the direction by which they exist, in that influence cannot be passed between nodes, say from a root node to the child of one of its parents [21]. This is a powerful measurement tool that allows for influence to be trickled up into the most important nodes, and would be beneficial in the context of this research to look for users that degree centrality may not rank accordingly based on weight and the flow of influence.

IV. Expiremental Results

A. Final Dataset

To reiterate from a previous section on the contents of the dataset collected for this research, initial collection began with 1,000 users that were selected based on mentions or combinations of pre-selected keywords relating to the COVID-19 pandemic. Through these 1,000 users, a final dataset encompassing 10,294 nodes connected by 20,590 edges was created. These edges were weighted on a scale of 3 to 1, which is inversely proportional to the number of jumps require to get to each subsequent user through their edges. A backlog of each vertex and edge in the network was kept for each step of the selection process in case more features or connections were desired at some point in the experimental phase. Additionally, to protect the privacy of all users that unknowingly contributed to this research, all user ID's that would be visible in final visuals were encoded from the ID format into integer values to prevent outside parties from interacting with them through this work. This is the only use of preprocessing that was utilized.

B. Methodology Explained

For each metric that is evaluated within this section a separate set of code was created due to the amount of processing time it would take to perform them all at once. While such an endeavor might be possible on a commercial scale; which would likely be where these methods would find traction, our use of this data was not hindered a great deal by separate code executions. The general process for which all clustering methods were progressed through is as follows:

a) Conversion of dataset into adjacency matrix.

b) Visualization to deduce cluster count (Eigenvalues, Dendrogram, WCSS).

c) Adjacency matrix normalized before principal component analysis for decomposition.

d) Decomposition of normalized dataset into two-dimensions for plotting and label acquisition.

e) Clustering applied through Sci-Kit: Learn [9].

f) Principal calculated through decomposition plotted using matplotlib [8] with cluster labels for each coordinate point.

Each of these steps were followed through a minimum of five times, which allowed numerous different visuals of the post-clustered networks to be picked from. Cluster mappings were fairly consistent through each iteration, and the ones that appear to have the most correlation to one another were selected for exhibition.

For centrality measurements, a simpler process was utilized since the measurement algorithms are contained within NetworkX [7]. An encoded edge list csv file with the user ID's of each user replaced with a unique number was used to create a NetworkX [7] graph instance, that was then inputted as a parameter to three different centrality functions. These functions return a dictionary object, which was then sorted by centrality score to bring the top five most central figures found through each method to the top. The key (ID number) and centrality score of each user were then combined into a string and inputted into corresponding columns within a Pandas library data frame [23]. This data frame was then exported to csv for storage on the disk as well as for visualization through a table.

C. Discussion of Results

1) Spectral Clustering Results

Before producing a visual output of the clusters for spectral clustering, a practice that involves the graphing of the first k eigenvalues that exist before the spectral gap was graphed and assessed. This is known as the eigengap heuristic and is validated through spectral graph theory [13]. Fig. 2 shows the usage of this tool, and since there exists 16 eigenvalues before the gap occurs, it can be deduced that there is likely 16 clusters.

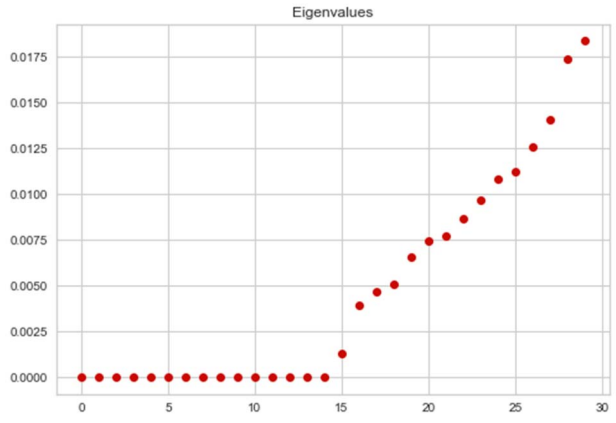

Fig. 2. Eigenvalue Heuristic Graph

After finding the estimated number of clusters, all that must be done next is to generate the feature vector of the graph from the Laplacian to find the labels for each of the points. The normal variant of K-means was then used to find this labeling, however the K-means++ algorithm would have been sufficient as well. Fig. 3 showcases the resulting distribution of clusters.

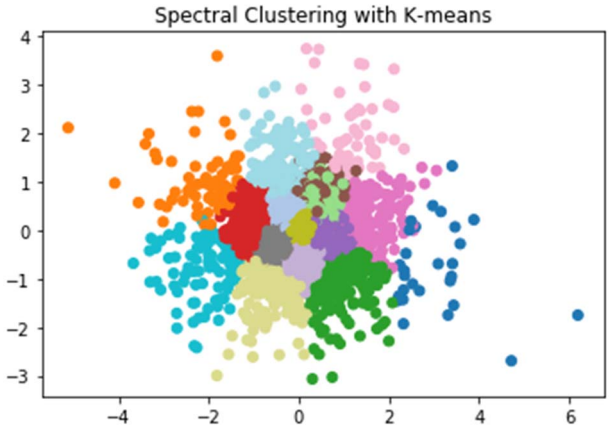

Fig. 3. Spectral clustering graph using K-means

2) Agglomerative Hierarichal Clustering Results

In similar practice to the process described within the above spectral clustering results, before any clustering began a dendrogram visual was first created to acquire a basic idea of the estimated number of clusters. The dendrogram shows the hierarchical structure of the clusters, and while it is not a perfect measure of revealing the number of clusters it does allow for some distinction to be made in finding the cluster count. Fig. 4 shows the visual of this dendrogram, which was created using the SciPy library [22].

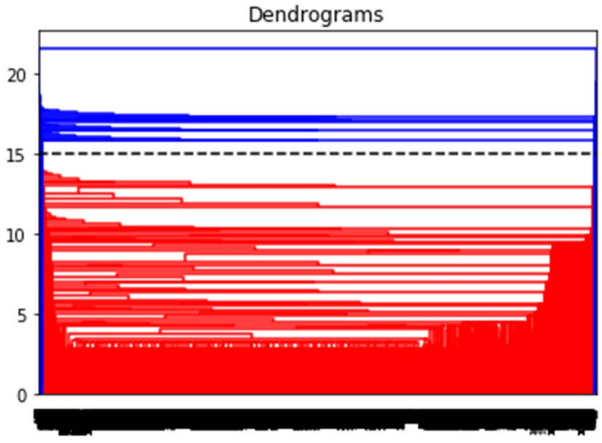

Fig. 4. Dendrogram model of data

By looking for the largest gap between partitions of the hierarchies, a guess can be made at how many clusters are likely to be within this data. The black dotted line serves to mark this gap, and from this it can be assumed that there exists 15 clusters through this method. Passing this parameter into the corresponding clustering class object, Fig. 5 is the resulting graph that is returned.

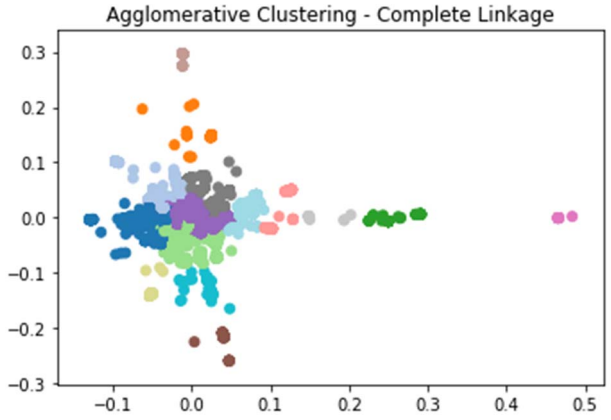

Fig. 5. Agglomerative hierarchical clustering with complete linkage graph

While a large amount of points within this graph could be considered noise by using this method, the distribution of points about the center shows promise that there is indeed several groupings of users that are found to be connected, and that the algorithm was able to pick them out accordingly despite their close proximity. Recall that the linkage method utilized is complete, which considers the dissimilarity between each of the clusters for assigning labels [15]. This shows us that the users within each of the clusters have some significant difference with the users of other clusters.

3) K-means Results

Once again, before beginning any clustering, a visual capable of depicting the number of clusters is first evaluated to determine the value to pass to the clustering class . For K-means, the most common visualization that shows the number of clusters can be found in graphing the within cluster sum of squares (WCSS) against a number of *k* clusters [24]. Another name for this graph is "The Elbow Method," and is named as such because the point that looks like the elbow of the curve is taken to be the best fitting number of clusters. [24] Fig. 6 demonstrates the elbow method while using the K-means++ method on the data.

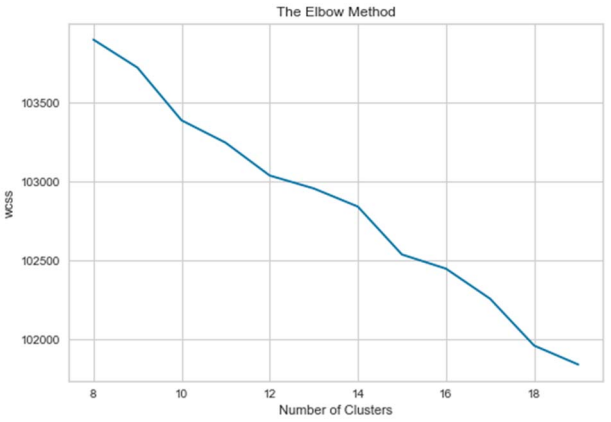

Fig. 6. The Elbow Method graph (*WCSS vs. k-clusters*)

Although this elbow is slightly distorted, likely due to the high ratio of WCSS to *k* cluster count, it would seem that the most logical "elbow" of this graph would be located at *k*=15 clusters. This also shows correlation to the previous clustering algorithms with spectral clustering at 16 clusters found via the graphing of the eigenvalues, and hierarchical clustering at 15 clusters found through the dendrogram. This number of clusters was then passed to the K-means clustering model instance along with the initialization parameter "K-means++," to produce Fig. 7.

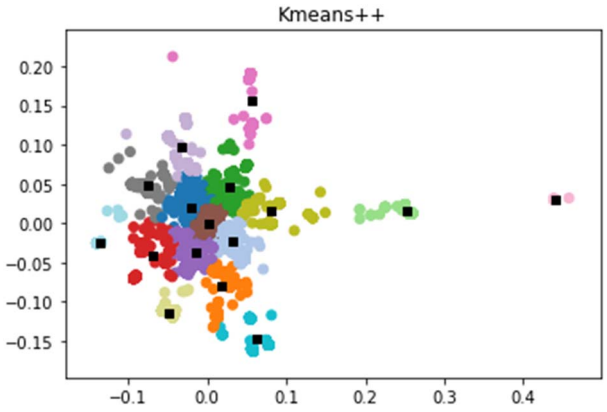

Fig. 7. K-means++ clustering graph

The distribution of this graph shows similar form to the one produced by the hierarchical clustering discussed earlier, although with far less points which could be considered noise. Since K-means is a centroid based clustering algorithm, it is a simple task to plot the centroids for each of the clusters for better visualization. These centroids also provide the added benefit of allowing for the MSE to be calculating by summing the squared difference between each point within a cluster with its center and dividing it by the total number of items in the graph. For the sake of consistency, five values of MSE from different program executions were then averaged to find an MSE with a value of 0.00035. Given that the MSE would always decrease for an infinitely many number of clusters since the points would eventually converge at their centroids, by balancing it against the number of clusters it has usage as another validity check for determining the most logical number of clusters.

D. Centrality Results

Collected centrality scores and their associated nodes for each of the three methods were collated together and stored as a csv file. The three centrality measures we experimented with in this research were closeness, degree, and PageRank. Table 1 shows the top five nodes with the highest centrality score (rounded to the fifth decimal place) for each given method.

TABLE I. CENTRALITY SCORES

User (Centrality Score)		
Closeness	***Degree***	***PageRank***
26 (0.1615)	8 (0.0112)	8 (0.00194)
54 (0.1557)	15 (0.011)	62 (0.00189)
13 (0.154)	10 (0.011)	21 (0.00189)
43 (0.1533)	56 (0.011)	67(0.00188)
21 (0.1524)	97 (0.0109)	56 (0.00187)

Comparing these different measurements, we find that for each method used there are many different nodes considered, as well ones that make repeated appearances. Based on the score ranking, it can be deduced that node 26 has the most

neighboring users surrounding it, whilst node eight has both the highest number of connections in the network as well as the highest influence through the connections of its child nodes. In similar fashion for the other used methods, these measures were collected five times to check for consistency. No changes can be reported between each of the run times.

V. Conclusion

In this research we have examined the usage of some common unsupervised machine learning practices as a means of aiding health surveillance. By locating the key figures in a network and discovering the clusters they exist in, this application would allow for a more direct prediction of the way a disease is able to spread through a community, and in the case of this work, how Covid-19 locally affects the population.

The resulting conclusions that we have been able to draw from using these algorithms on this dataset is that there is an average of 15 groups that exist within the overall scheme of the network. Several key figures were also reported through calculating centrality, and further analysis of this dataset would likely examine these individuals and their connections even further. Given the current methodology that was used, it would not be a difficult task to repurpose existing code to expand the network even further through these key figures.

Plans for the future may see the ideology of this work be applied to real world social connections, which of course would be much more noteworthy in viral disease control but require more complicated methods. Another such use may allow these methods to stand on their own by linking social and medical outlets of patients to determine their associated risk of infection through their social groups, or to classify some individuals as potential super spreaders. Whatever the case may be, machine learning undoubtedly has a future in the health surveillance system, and as its most potent applications in the field are discovered, society will certainly benefit.

Acknowledgment

This work was sponsored by a research grant from Morehead State University.

References

[1] Cheng, C. K., Lau, E. H., Ip, D. K., Yeung, A. S., Ho, L. M., & Cowling, B. J. (2009). A profile of the online dissemination of national influenza surveillance data. BMC Public Health, 9(1). doi:10.1186/1471-2458-9-339

[2] Haston, J., & Pickering, L. (2020, August 29). CDC's disease surveillance system critical for public health. Retrieved from https://www.aappublications.org/news/2019/03/08/mmwr030819

[3] Phalkey, R. K., Yamamoto, S., Awate, P., & Marx, M. (2013). Challenges with the implementation of an Integrated Disease Surveillance and Response (IDSR) system: Systematic review of the lessons learned. Health Policy and Planning, 30(1), 131-143. doi:10.1093/heapol/czt097

[4] Chae, S., Kwon, S., & Lee, D. (2018). Predicting Infectious Disease Using Deep Learning and Big Data. International Journal of Environmental Research and Public Health, 15(8), 1596. doi:10.3390/ijerph15081596

[5] Shin, S., Seo, D., An, J., Kwak, H., Kim, S., Gwack, J., & Jo, M. (2016). High correlation of Middle East respiratory syndrome spread with Google search and Twitter trends in Korea. Scientific Reports, 6(1). doi:10.1038/srep32920

[6] Roesslein, J. (2009). tweepy Documentation. Online] http://tweepy. readthedocs. io/en/v3, 5.

[7] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, "Exploring network structure, dynamics, and function using NetworkX", in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008

[8] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[9] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[10] Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. Annals of Data Science, 2(2), 165-193. doi:10.1007/s40745-015-0040-1

[11] Rui Xu and D. Wunsch, "Survey of clustering algorithms," in IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645-678, May 2005, doi: 10.1109/TNN.2005.845141.

[12] Bach, F. R., & Jordan, M. I. (2004). Learning spectral clustering. In Advances in neural information processing systems (pp. 305-312).

[13] Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing, 17(4), 395-416.

[14] Berkhin, P. (2006). A survey of clustering data mining techniques. In Grouping multidimensional data (pp. 25-71). Springer, Berlin, Heidelberg.

[15] Murtagh, F., & Contreras, P. (2011). Methods of hierarchical clustering. arXiv preprint arXiv:1105.0121.

[16] Uppada, S. K. (2014). Centroid based clustering algorithms—A clarion study. International Journal of Computer Science and Information Technologies, 5(6), 7309-7313.

[17] Arthur, D., & Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Stanford.

[18] Qin Wu, Xingqin Qi, Eddie Fuller, Cun-Quan Zhang, "“Follow the Leader”: A Centrality Guided Clustering and Its Application to Social Network Analysis", The Scientific World Journal, vol. 2013, Article ID 368568, 9 pages, 2013. https://doi.org/10.1155/2013/368568

[19] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. 2014. Computing classic closeness centrality, at scale. In Proceedings of the second ACM conference on Online social networks (COSN '14). Association for Computing Machinery, New York, NY, USA, 37–50. DOI:https://doi.org/10.1145/2660460.2660465

[20] Heidemann, J., Klier, M., & Probst, F. (2010). Identifying key users in online social networks: A pagerank based approach.

[21] Disney, A. (2020, January 2). Social network analysis 101: Centrality measures explained. Retrieved from https://cambridge-intelligence.com/use-cases/social-networks/

[22] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, in press.

[23] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

[24] Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and k-means in WSN. International Journal of Computer Applications, 105(9).

[25] Dion, M., AbdelMalik, P., & Mawudeku, A. (2015). Big Data and the Global Public Health Intelligence Network (GPHIN). Canada communicable disease report = Releve des maladies transmissibles au Canada, 41(9), 209–214. https://doi.org/10.14745/ccdr.v41i09a02