Comparison of dimensionality reduction and clustering methods for SARS-CoV-2 genome

Untari N. Wisesty, Tati Rajab Mengko
School of Electrical and Information Engineering, Bandung Institute of Technology, Indonesia

Article Info

Article history:

Received Jan 13, 2021
Revised Apr 29, 2021
Accepted May 20, 2021

Keywords:

Autoencoder
Dimensionality reduction
Genome clustering
Principal component analysis
SARS-CoV-2

ABSTRACT

This paper aims to conduct an analysis of the SARS-CoV-2 genome variation was carried out by comparing the results of genome clustering using several clustering algorithms and distribution of sequence in each cluster. The clustering algorithms used are K-means, Gaussian mixture models, agglomerative hierarchical clustering, mean-shift clustering, and DBSCAN. However, the clustering algorithm has a weakness in grouping data that has very high dimensions such as genome data, so that a dimensional reduction process is needed. In this research, dimensionality reduction was carried out using principal component analysis (PCA) and autoencoder method with three models that produce 2, 10, and 50 features. The main contributions achieved were the dimensional reduction and clustering scheme of SARS-CoV-2 sequence data and the performance analysis of each experiment on each scheme and hyper parameters for each method. Based on the results of experiments conducted, PCA and DBSCAN algorithm achieve the highest silhouette score of 0.8770 with three clusters when using two features. However, dimensionality reduction using autoencoder need more iterations to converge. On the testing process with Indonesian sequence data, more than half of them enter one cluster and the rest are distributed in the other two clusters.

Corresponding Author:

Untari N. Wisesty
School of Electrical and Information Engineering
Bandung Institute of Technology
Bandung, Indonesia
Email: untarinw@telkomuniversity.ac.id

1. INTRODUCTION

In December 2019, a pneumonia outbreak pneumonia occurred in Wuhan which was caused by a new virus called covid-19 (coronavirus disease 2019) or SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2). The virus was first detected in Wuhan City, China, on 12 December 2019 [1]. The virus has a very high transmission rate, so that in just two months, the virus can spread from Wuhan to all of China, as well as 33 other countries. As of November 30, 2020, there were 62,363,527 confirmed cases of covid-19 and 1,456,687 deaths due to covid-19 worldwide [2]. In Indonesia, the first case of covid-19 was announced on March 2, 2020, and as of November 30, 2020, 534,266 positive cases have been detected and 16,815 people have died due to covid-19 [3]. Symptoms that often appear in covid-19 patients include fever, headache, cough, expectoration, fatigue, and dyspnea. However, some patients show symptoms of shortness of breath, haemoptysis, diarrhea, and there are even patients who do not show any symptoms at all [4].

The SARS-Cov-2 virus has a high mutation rate because it is included in viral RNA so that the variety of the virus increases [5], [6]. Therefore, analysis of variations in the viral genome is urgently needed.

Variations in the covid-19 virus can be detected through genome analysis from samples of infected patients. The samples are collected from parts of the body where the coronavirus collects, such as a person's nose or throat. These samples were inserted into the polymerase chain reaction (PCR) tool and used optimal primers and probes to extract RNA from the virus [7], [8]. Based on the extraction/sequencing results, the SARS-CoV-2 genome has a length of 29,891 bp with a frequency of G and C of 38% [9]. In terms of genetic structure, SARS-CoV-2 has a different genetic structure compared to the previously identified SARS-CoV and MERS-CoV viruses, with a similarity rate of 79% against SARS-CoV and nearly 50% against MERS-CoV [10].

In this paper, dimensionality reduction and clustering analyses were carried out to cluster genome data from patients with covid-19 indication to analyze virus variation in several country in Asia and Indonesia. The method used is the principal component analysis (PCA) and autoencoder for dimensionality reduction and several clustering algorithms for genome clustering. Dimensionality reduction is needed to reduce dimensions and remove redundant features [11] from genome data which has very high dimensions, so that the data can be processed in the clustering method. PCA reduces the dimension of the data by calculating the eigen value and eigen vector from the training data [12]. PCA has been applied to a wide variety of data including microarray data [13] and DNA sequence data [14]. Autoencoder is an effective method to reduce dimensions of image data, text, and other complex data [11], [15]-[18]. Then several clustering methods are used to cluster the data resulting from dimensionality reduction method, including K-means, Agglomerative hierarchical clustering (AHC), Gaussian mixture models (GMM), density-based spatial clustering of applications with noise (DBSCAN), and mean shift clustering. Furthermore, clustering performance is measured using a Silhouette score, which has a value in the range of -1 to 1.

The findings of this research are the dimensional reduction and clustering scheme of SARS-CoV-2 sequence data, the performance analysis of each experiment on each scheme and hyper parameters for each method, distribution of SARS-CoV-2 genome sample variations in China, Japan, South Korea, Singapore, and India for training data and Indonesia for testing data on each of the resulting clusters. Finally, this paper is arranged in the following order: section two (proposed method) about the proposed schema methodology, which includes data acquisition and preprocessing, PCA and autoencoder, and clustering algorithms; section three (results and discussion) about experimental scenarios, presentation of results and discussion; and the final section (conclusion) about the research conclusions.

2. RESEARCH METHOD

Figure 1 shows a diagram process of the SARS-CoV-2 sequence clustering system. Genome sequence data of patients infected with covid-19 were obtained from EpiCoV, GISAID [19] which amounted to 4935 data (access on 21 November 2020). The data taken comes from several countries in Asia, including China, Japan, South Korea, Singapore, India, and Indonesia. Genome sequence from China, Japan, South Korea, Singapore, and India will be training data, and genome sequence from Indonesia will be testing data. The data is stored in .fasta format so that it requires a parsing process for the acquisition of sequence data only, without other information. Raw data of SARS-CoV-2 sequence can be seen at Figure 2.

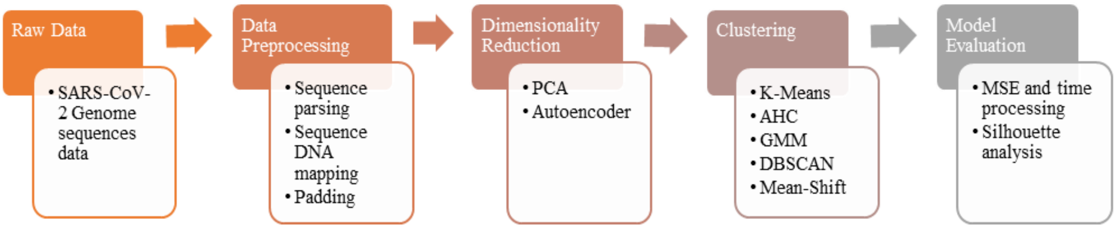
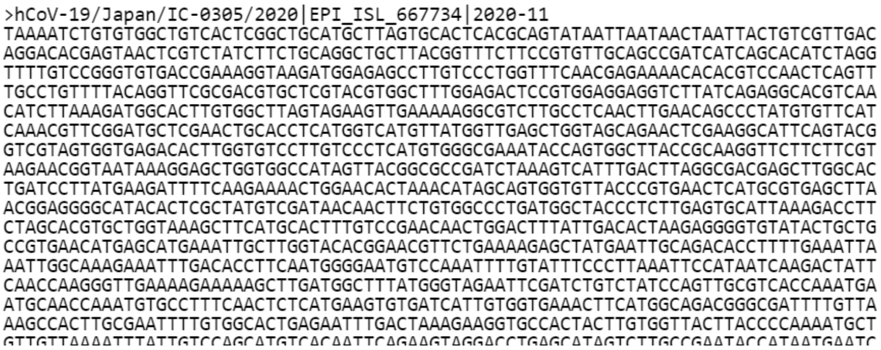


Figure 1. Pipeline of SARS-CoV-2 sequence clustering system

In addition to the parsing process, data preprocessing was also carried out to form a complete sequence with a size of 1xN, where N is the number of nucleotides in one data, because the raw sequence in the fasta file consists of several nucleotide lines. Mapping DNA sequences is needed to convert DNA sequences into numerical sequences so that they can be processed at a later stage. The mapping technique used is the integer representation [20], [21], by modifying the integer numbers assigned to the nucleotide A, C, T, and G, because the number 0 is used for the padding process and other values if there is a missing value. The equation used for sequence mapping is presented in (1). Then the padding process is carried out with the number 0 so that the length of the sequence becomes the same and produces a sequence length of 30,018. Padding process is needed because PCA and autoencoder requires the same input length for all data.



```
>hCoV-19/Japan/IC-0305/2020|EPI_ISL_667734|2020-11
TAAAATCTGTGTGGCTGTCACTCGGCTGCATGCTTAGTGCACTCACGCAGTATAATTAATAACTAATTACTGTCGTTGAC
AGGACACGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCTTCCGTGTTGCAGCCGATCATCAGCACATCTAGG
TTTTGTCCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTTGTCCCTGGTTTCAACGAGAAAACACACGTCCAACTCAGTT
TGCCTGTTTTACAGGTTCGCGACGTGCTCGTACGTGGCTTTGGAGACTCCGTGGAGGAGGTCTTATCAGAGGCACGTCAA
CATCTTAAAGATGGCACTTGTGGCTTAGTAGAAGTTGAAAAAGGCGTCTTGCCTCAACTTGAACAGCCCTATGTGTTCAT
CAAACGTTCGGATGCTCGAACTGCACCTCATGGTCATGTTATGGTTGAGCTGGTAGCAGAACTCGAAGGCATTCAGTACG
GTCGTAGTGGTGAGACACTTGGTGTCCTTGTCCCTCATGTGGGCGAAATACCAGTGGCTTACCGCAAGGTTCTTCTTCGT
AAGAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTTGACTTAGGCGACGAGCTTGGCAC
TGATCCTTATGAAGATTTTCAAGAAAACTGGAACACTAAACATAGCAGTGGTGTTACCCGTGAACTCATGCGTGAGCTTA
ACGGAGGGGCATACACTCGCTATGTCGATAACAACTTCTGTGGCCCTGATGGCTACCCTCTTGAGTGCATTAAAGACCTT
CTAGCACGTGCTGGTAAAGCTTCATGCACTTTGTCCGAACAACTGGACTTTATTGACACTAAGAGGGGTGTATACTGCTG
CCGTGAACATGAGCATGAAATTGCTTGGTACACGGAACGTTCTGAAAAGAGCTATGAATTGCAGACACCTTTTGAAATTA
AATTGGCAAAGAAATTTGACACCTTCAATGGGGAATGTCCAAATTTTGTATTTCCCTTAAATTCCATAATCAAGACTATT
CAACCAAGGGTTGAAAAGAAAAAGCTTGATGGCTTTATGGGTAGAATTCGATCTGTCTATCCAGTTGCGTCACCAAATGA
ATGCAACCAAATGTGCCTTTCAACTCTCATGAAGTGTGATCATTGTGGTGAAACTTCATGGCAGACGGGCGATTTTGTTA
AAGCCACTTGCGAATTTTGTGGCACTGAGAATTTGACTAAAGAAGGTGCCACTACTTGTGGTTACTTACCCCAAAATGCT
GTTGTTAAAATTTATTGTCCAGCATGTCACAATTCAGAAGTAGGACCTGAGCATAGTCTTGCCGAATACCATAATGAATC
```

Figure 2. Raw data of SARS-CoV-2 sequences

$$X'(i) = \begin{cases} 1, & X(i) = T \\ 2, & X(i) = C \\ 3, & X(i) = A \\ 4, & X(i) = G \end{cases} \tag{1}$$

To extract important features in sequence data and reduce dimension data, feature extraction/dimensionality reduction is performed using the PCA and autoencoder method as shown in Figure 3. PCA reduces data dimension by transforming it into principal component obtained from calculating the eigenvalues and eigenvectors of training data. The eigenvectors that have the highest eigenvalues generated by PCA represent the most important features [22]. The eigenvalues and eigenvectors are obtained by solving the (2), where C_X is covariance matrix of input X, v_m is eigenvectors, and λ_m is eigenvalues [13]. The eigenvectors are sorted based on their eigenvalues which then become the principal components. In this paper, three PC models from the PCA reduction dimensions are compared which produce data with 50 dimensions (PCA_50), 10 dimensions (PCA_10), and 2 dimensions (PCA_2).

$$C_X v_m = \lambda_m v_m \tag{2}$$

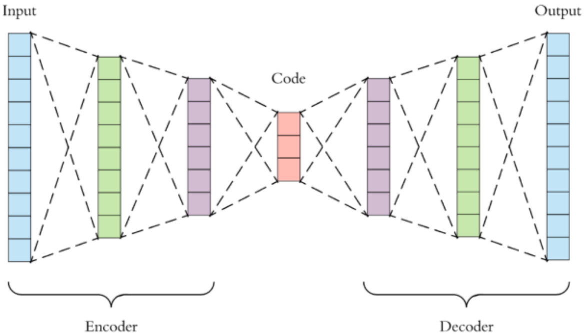


Figure 3. General autoencoder architecture [17]

The autoencoder model that is built consists of encoder and decoder. The encoder functions to extract features from the data used and also to reduce the dimensions of the data, while the decoder is used to reconstruct the reduced data into initial data [23], [24]. There are three architectures of the autoencoder model used in this task as shown in Table 1, which have differences in the depth of the architecture and the number of output features extracted. The first autoencoder architecture (AE_50) use four dense layers (fully connected layer) for model encoder which have 500, 250, 100, and 50 neurons for each layer, and four dense layers for model decoder (100, 250, 500, and 30,018 neurons for each layer). The second autoencoder architecture (AE_10) use five dense layers for model encoder (500, 250, 100, 50, and 10 neurons for each layer), and five dense layers for model decoder (50, 100, 250, 500, and 30,018 neurons for each layer). The third autoencoder architecture (AE_2) use five dense layers for model encoder (500, 250, 100, 50, and 2 neurons for each layer), and five dense layers for model decoder (50, 100, 250, 500, and 30,018 neurons for each layer). These three models are trained with Adam's optimization algorithm [25], [26] which is a first order gradient descent optimization algorithm and estimates the adaptive learning rate based on lower order moments. This algorithm is computationally efficient and requires little memory, so it is suitable for problems with large amounts of data or parameters. Loss function used is the mean square error (MSE).

Table 1. Autoencoder architecture model for dimensionality reduction of SARS-CoV-2 sequence.

Model	AUTOENCODER 1 (AE_50)			AUTOENCODER 2 (AE_10)			AUTOENCODER 3 (AE_2)		
	Layer Type	Output	# Param	Layer Type	Output	# Param	Layer Type	Output	# Param
Encoder:	Input	30018	-	Input	30018	-	Input	30018	-
	Dense	500	15,009,500	Dense	500	15,009,500	Dense	500	15,009,500
	Dense	250	125,250	Dense	250	125,250	Dense	250	125,250
	Dense	100	25,100	Dense	100	25,100	Dense	100	25,100
	Dense	50	5,050	Dense	50	5,050	Dense	50	5,050
				Dense	10	510	Dense	2	102
	Total Param:		15,164,900	Total Param:		15,165,410	Total Param:		15,165,002
Decoder:	Input	50	-	Input	10	-	Input	2	-
	Dense	100	5,100	Dense	50	550	Dense	50	150
	Dense	250	25,250	Dense	100	5,100	Dense	100	5,100
	Dense	500	125,500	Dense	250	25,250	Dense	250	25,250
	Dense	30018	15,039,018	Dense	500	125,500	Dense	500	125,500
				Dense	30018	15,039,018	Dense	30018	15,039,018
	Total Param:		15,194,868	Total Param:		15,195,418	Total Param:		15,195,018
Autoencoder:	Input	30018	-	Input	29946	-	Input	29946	-
	Encoder	50	15,164,900	Encoder	50	15,165,410	Encoder	100	15,165,002
	Decoder	30018	15,194,868	Decoder	29946	15,195,418	Decoder	29946	15,195,018
	Total Param:		30,359,768	Total Param:		30,360,828	Total Param:		30,360,020

In this paper, we also compare the results of clustering the sequence data resulting from dimensional reduction using several clustering methods, including K-means, AHC, GMM, DBSCAN, and mean shift clustering. The K-means algorithm starts by randomly initializing the centroid points according to a predetermined number of clusters [27], [28]. Then calculate the distance of each data to each centroid to determine the cluster for each data point. The update of the centroid position is done by calculating the average value of the data points contained in each cluster. These processes are carried out until the positions of the centroid has not changed or the maximum number of iterations is reached.

GMM is more flexible than K-means in formed clusters. In GMM, there are two parameters that are calculated to update the centroid point, namely the average value and standard deviation, so that the cluster formed can be various kinds of ellipses. In calculating the average value and standard deviation, GMM uses an optimization algorithm, namely expectation-maximization (EM) and Gaussian distribution for each cluster [29], [30]. AHC uses a bottom up algorithm where each data point is considered as one cluster and combines the clusters that have the closest distance [31]. Distance calculations can be done by calculating the average value of data points in a cluster or by calculating the shortest distance between data points in one cluster and data points in another cluster. This process is carried out until one large cluster is obtained and then the desired number of clusters is selected.

The mean-shift clustering algorithm begins by randomly determining the data points that are the center points of the sliding window with a certain radius [32]. The sliding window will shift towards an area with a higher density level (number of data points in the window) until convergent conditions are reached. Mean-shift clustering has the advantage that the number of clusters does not need to be determined in advance because the algorithm can find the optimal number of clusters automatically. DBSCAN is an algorithm similar to mean-shift clustering that can determine the number of clusters automatically. DBSCAN starts by selecting a data point that has never been visited, then other data points adjacent to the data point within a threshold distance will be included in the same cluster [33]. If the number of data points in the cluster is more than the minPoints parameter (the minimum number of data points in a cluster), then the data points will become one cluster and marked as visited. The process will repeat until there are no more data points that can be visited. Data points that do not include to any cluster are considered noise.

Dimensionality reduction data using three autoencoder models and PCA will be the input for the clustering algorithm, which has 2, 10, and 50 features. For K-means, AHC, and GMM algorithms, the number of clusters is observed from 2 to 50 clusters. Meanwhile, for the mean-shift clustering algorithm and DBSCAN, the number of clusters does not need to be determined at the beginning. To evaluate the performance of the clustering algorithm, Silhouette analysis is used based on the Silhouette Score [34]. The silhouette score determines the degree of separation between clusters which has a value in the range [-1,1], where the value 1 indicates the sample data is very far from the other cluster, the value of 0 indicates the sample data is very close to the other cluster, and the value -1 indicates the sample data is included in the wrong cluster.

3. RESULTS AND DISCUSSION

In this paper a system was built to reduce dimension and cluster genome data form SARS-CoV-2 virus to analize the genome virus variation. The method used is PCA and autoencoder for dimensional reduction, and several clustering algorithms which includes K-means, Gaussian mixture models (GMM), agglomerative hierarchical clustering (AHC), density-based spatial clustering of applications with noise (DBSCAN), and mean shift clustering. To test the system performance that has been built, experiment is done by comparing silhouette score of clustering methods for various input dimension from PCA and autoencoder for training and validation process. As for the testing process, the genome sequence from Indonesia was included in the dimension reduction and clustering model that was previously built to analyse the variation of viruses found in Indonesia. Following are the results and analysis of the experiments that have been carried out.

3.1. Autoencoder performance for dimensionality reduction

This paper proposed three model autoencoder for dimensionality reduction, namely AE_50, AE_10, and AE_2, that produce data with 50, 10, and 2 dimensions, respectively. Based on the results presented in Table 2, the MSE training and validation of AE_50 and AE_10 is better than the AE_2. This shows that the AE_50 and AE_10 converge faster than the autoencoder 1. For AE_2, the dimensionality reduction process that occurs is very large, so the reconstruction process into original data is still not good. The training and validation running time does not have a significant difference because there is only a slight difference in the number of parameters/weights of the autoencoder architecture of the three models.

Table 2. MSE and running time comparison of training and validation processes for autoencoder models 1, 2, and 3

Model	MSE		Running Time	
	Training	Validation	Training	Validation
AE_50	0.101	0.098	2 s/epoch; 12 ms/step	1 s/epoch; 6 ms/step
AE_10	0.1503	0.1531	2 s/epoch; 13 ms/step	1 s/epoch; 7 ms/step
AE_2	0.7409	0.7527	2 s/epoch; 12 ms/step	1 s/epoch; 6 ms/step

3.2. Number of clusters observation on K-means, GMM, and AHC

The number of cluster effect parameters of the K-Means, GMM, and AHC clustering algorithms based on silhouette score of clustering results of the features generated by the PCA and autoencoder models. PCA and autoencoder model that was built produced data with 2, 10, and 50 features. The parameters used in the K-Means, GMM, and AHC algorithms were the number of clusters, and in this study the number of clusters was observed in the range of 2 to 50 clusters. K-means, AHC, and GMM are included in clustering-based algorithms so that the selection of the number of clusters and the initial initialization of centroids greatly affect the results of clustering [35]. Based on the results obtained in Figure 4 to Figure 6, it can be concluded that the output of the PCA_2 with 2 features have the highest silhouette score using the K-means, GMM, and AHC algorithms, for all the number of clusters 2 to 50. However, AE_2 with 2 features have more stable silhouette score over all number of clusters. Data with 10 and 50 features have smaller silhouette score relatively. This can happen because of the clustering algorithm is more able to cluster fewer features so that the data contain less noise or redundancy. Based on the results of effect of the number of clusters observations, the greater the number of clusters, the higher the Silhouette score obtained for data with 10 and 50 features.

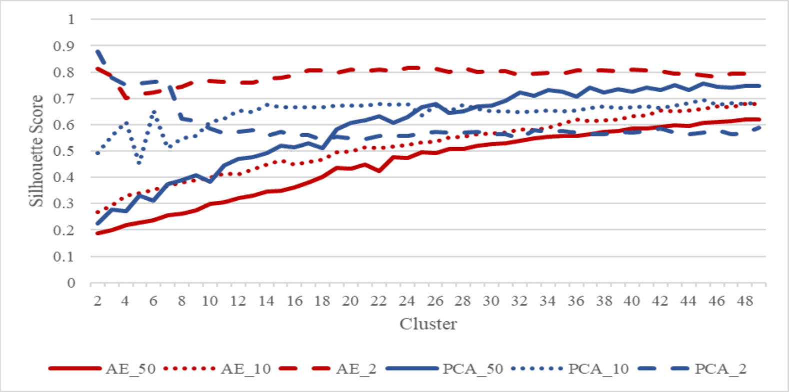


Figure 4. Silhouette score of K-means clustering results

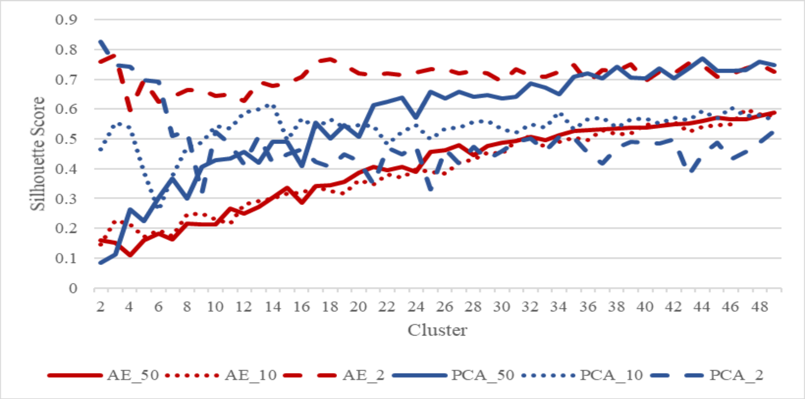


Figure 5. Silhouette score of GMM clustering results

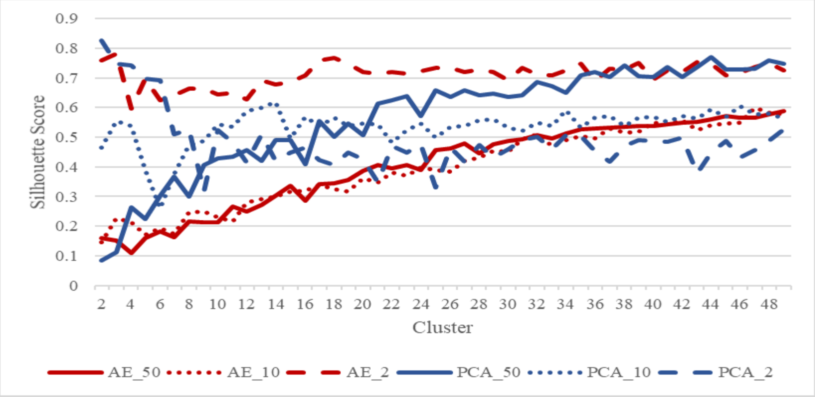


Figure 6. Silhouette score of AHC clustering results

3.3. Quantile parameter observation on mean-shift clustering

The quantile parameter is used to calculate the bandwidth (window size/radius) in the mean-shift clustering algorithm, by calculating the median pairwise distance of the sample used. The value of the quantile parameter ranges in the range 0 to 1, value of 0.5 means that the median value of all pairwise distances is used. In this study, the observed quantile values were 0.01, 0.05, 0.1, 0.15, 0.2, and 0.25. Based on Table 3 and Table 4, the greater the quantile value, the bigger the resulting window size/bandwidth, so that the number of clusters produced is smaller. A large quantile value also results a higher silhouette score for data with two features, but this is not the case for data with 10 and 50 features.

Table 3. Estimated number of clusters of mean-shift clustering results

Quantile	Estimated Number of Clusters					
	AE_50	AE_10	AE_2	PCA_50	PCA_10	PCA_2
0.01	**564**	**367**	96	17	354	120
0.05	26	40	29	**35**	24	53
0.1	-	4	5	2	11	**5**
0.15	-	3	5	-	**10**	**5**
0.2	-	2	**4**	-	**10**	**5**
0.25	-	2	4	-	9	**5**

Table 4. silhouette score of mean-shift clustering results

Quantile	Silhouette Score					
	AE_50	AE_10	AE_2	PCA_50	PCA_10	PCA_2
0.01	**0.738**	**0.799**	0.771	0.302	0.67	0.546
0.05	0.376	0.532	0.773	**0.673**	0.545	0.485
0.1	-	0.229	0.776	0.155	0.701	**0.783**
0.15	-	0.235	0.776	-	**0.708**	**0.783**
0.2	-	0.381	**0.784**	-	**0.708**	**0.783**
0.25	-	0.368	0.779	-	0.703	**0.783**

3.4. Epsilon parameter observation on DBSCAN clustering

The epsilon parameter is a parameter used to determine whether a data point belongs to the same cluster or not in DBSCAN. A data point can enter the same cluster if the distance is less than epsilon. In this study, the observed epsilon values were 0.5, 1, 5, 10, 20, 30, 40, and 50. Based on Table 5 to Table 7, it can be concluded that each different data will require a certain epsilon value that is different depending on the distribution of the data. However, data with less features will need smaller epsilon value. DBSCAN can also detect outlier data, where the data is very far from the existing cluster, rather than forcing the outlier data to enter a specific cluster.

Table 5. Estimated number of clusters of DBSCAN clustering results

Eps	Estimated number of clusters					
	AE_50	AE_10	AE_2	PCA_50	PCA_10	PCA_2
0.5	106	74	26	99	87	40
1	99	72	13	87	76	23
5	79	88	5	73	64	11
10	88	**95**	**4**	80	35	7
20	**94**	84	2	76	**15**	**3**
30	82	15	2	68	12	2
40	27	3	-	**60**	3	-
50	2	2	-	52	3	-

Table 6. Estimated number of noise points of DBSCAN clustering results

Eps	Estimated number of noise points					
	AE_50	AE_10	AE_2	PCA_50	PCA_10	PCA_2
0.5	2106	1535	40	2084	1637	502
1	1795	1425	16	1787	1443	245
5	1323	1120	0	1396	757	39
10	1173	672	0	1249	321	6
20	869	261	0	737	106	0
30	515	21	0	475	27	0
40	145	6	-	343	9	-
50	2	3	-	249	3	-

Table 7. Silhouette score of DBSCAN clustering results

Eps	Silhouette Score					
	AE_50	AE_10	AE_2	PCA_50	PCA_10	PCA_2
0.5	0.27	0.484	0.513	0.189	0.349	0.131
1	0.364	0.533	0.603	0.314	0.448	-0.122
5	0.566	0.592	0.682	0.525	0.548	0.692
10	0.605	**0.656**	**0.79**	0.568	0.393	0.649
20	**0.629**	0.614	0.732	0.636	**0.6**	**0.877**
30	0.548	-0.09	0.732	0.69	0.365	0.764
40	0.076	0.068	-	**0.715**	0.141	-
50	0.095	0.276	-	0.586	0.219	-

3.5. Comparison of five clustering algorithms results

In Table 8 and Table 9, in the number of features 10 and 50 from AE_50 and AE_10, mean-shift clustering has the best silhouette score (0.7380 and 0.784) compared to other algorithms, but it is likely not optimal because the estimated number of clusters really large (564 and 367 clusters). This is different for data with 50 features from PCA_50 which gets a Silhouette Score of 0.7791 from the results of the AHC clustering algorithm and the resulting number of clusters is 48 clusters. Meanwhile, for data with 10 features, PCA_10 achieved a silhouette score of 0.7080 and the number of clusters was 10 using the mean-shift Clustering algorithm. Data with 2 features has the best silhouette score, both for the reduction of autoencoder (AE_2) and PCA (PCA_2) dimensions, namely 0.8133 for AE_2 and 0.877 for PCA_2. The two reduced data can be clustered into 3 clusters using the AHC algorithm for AE_2 and DBSCAN for PCA_2. This proves that the dimension reduction process carried out is quite good because the training data is taken from five countries, namely China, South Korea, Singapore, Japan, and India that have the geographical proximity of the country.

DBSCAN and mean-shift clustering have the advantage of being able to determine the optimal number of clusters automatically, while in the K-means, GMM, and AHC algorithms the number of clusters must be initialized at the start. In terms of data clustering characteristics, AHC has similarities in the formation of phylogenetic trees, which construct trees by calculating the distance / similarity between sequences. DBSCAN can identify outliers in the data and consider them to be noise, whereas in Mean-shift clustering, the outliers are forced into a particular cluster. DBSCAN can have any shape and size of the cluster, while in Mean-shift, the cluster formed is circular. However, DBSCAN has the disadvantage of not having a good performance if the data has high dimensions because estimating the epsilon value becomes more difficult.

Table 8. Silhouette score comparison of clustering algorithms

Clustering Algorithm	Silhouette Score					
	AE_50	AE_10	AE_2	PCA_50	PCA_10	PCA_2
K-means	0.6196	0.6786	0.8180	0.7559	0.6939	0.8766
GMM	0.5882	0.6006	0.7808	0.7704	0.6190	0.8258
AHC	0.6139	0.6682	**0.8133**	**0.7791**	0.6778	0.8766
Mean-Shift	**0.7380**	**0.7990**	0.7840	0.6730	**0.7080**	0.7830
DBSCAN	0.6290	0.6560	0.7900	0.7150	0.6000	**0.8770**

Table 9. Optimal number of clusters comparison of clustering algorithms

Clustering Algorithm	Optimal Number of Clusters					
	AE_50	AE_10	AE_2	PCA_50	PCA_10	PCA_2
K-means	50	49	2	46	46	3
GMM	50	50	4	45	15	3
AHC	50	50	**3**	**48**	49	3
Mean-Shift	**564**	**367**	4	35	**10**	5
DBSCAN	94	95	4	60	15	**3**

3.6. Training and testing genome sequence in each cluster

Figure 7 and Figure 8 show the distribution of training data in each cluster based on the best silhouette score from the experiments that have been carried out, namely AE_2 data are clustered using the AHC algorithm and PCA_2 data are clustered using the DBSCAN algorithm. The training data used is SARS-CoV-2 sequence data from China, Japan, South Korea, Singapore, and India. In both figures, the sample distribution from each country in each cluster is almost the same. So, it can be concluded that Cluster 1, Cluster 2, and Cluster 3 in Figure 7 are the same as Cluster 3, Cluster 1, and Cluster 2 in Figure 8, respectively. In the data resulting from AE_2 and PCA_2, the sample data is spread out over one cluster. This can be due to the SARS-CoV-2 sequences in the five countries that have a high majority of similarities. The testing data used were SARS-CoV-2 sequences from Indonesia, totaling 109 sequences. Figure 9 shows the distribution of testing data in each cluster, which is the result of the dimensional reduction and clustering processes using the two best models obtained in the training process, namely AE_2 data are clustered using the AHC algorithm and PCA_2 data are clustered using the DBSCAN algorithm. These results show that the Indonesian sequence data has a slightly different distribution from the training data. However, half of the data is still gathered in one cluster such as training data, while the other data is spread over the other two clusters.

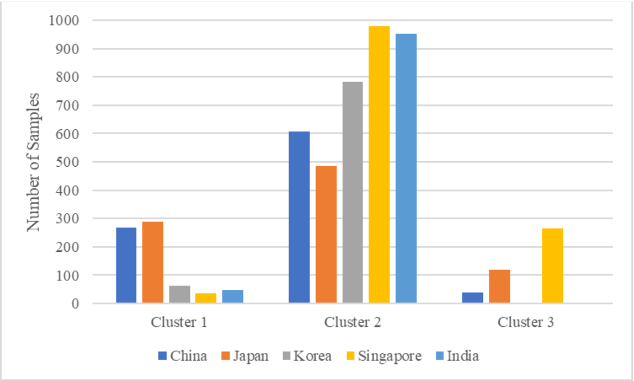


Figure 7. Distribution of training data in each cluster with the results of the AE_2 reduction and the AHC algorithm

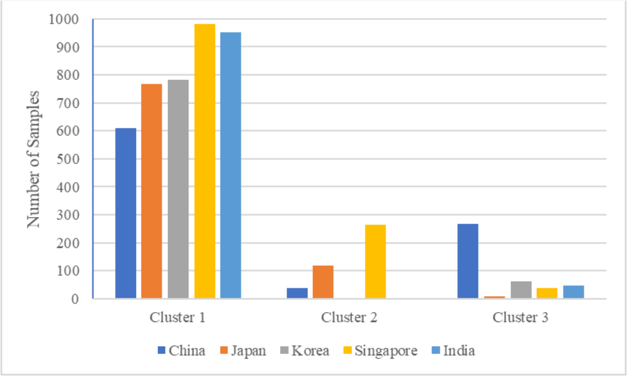


Figure 8. Distribution of training data in each cluster with the results of the PCA_2 reduction and the DBSCAN algorithm

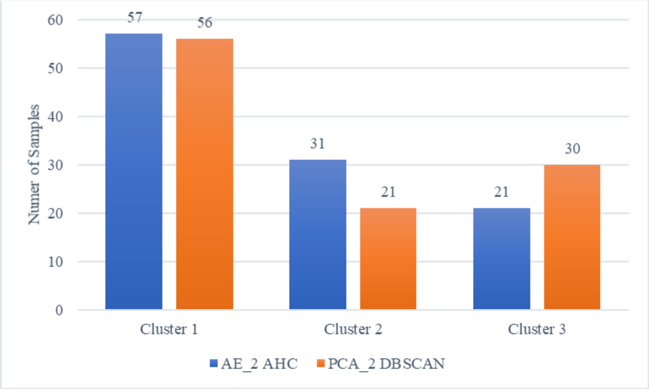


Figure 9. Distribution of testing data in each cluster

4. CONCLUSION

In this paper, a dimensional reduction and clustering system was developed for the genome data of the SARS-CoV-2 virus. Dimensional reduction is done by using PCA and autoencoder with three models that produce reduced data with 2, 10, and 50 features. The clustering algorithms used are K-means, GMM, AHC, Mean-shift clustering, and DBSCAN. Based on the experimental results, the system built can achieve the highest silhouette score of 0.8770 with three clusters when using two features of PCA and DBSCAN algorithm. As well as a silhouette score of 0.8133 with three clusters using two features of the autoencoder and the AHC algorithm. Meanwhile, the reduced sequences with features 10 and 50 have a smaller silhouette score in all clustering algorithms used. The K-means, GMM, and AHC algorithms require a predefined number of clusters. Meanwhile, mean-shift clustering and DBSCAN algorithms can find the optimal number of clusters automatically, so that it is an advantage when compared to the K-means, GMM, and AHC algorithms. The optimal number of clusters with the highest silhouette score consists of three clusters, either using data result from the reduction dimension of the autoencoder or PCA with two features.

This paper also analyses the distribution of training data and testing in each cluster formed using two models that have the best silhouette scores. The training data is a SARS-CoV-2 sequence from China, Japan, South Korea, Singapore, and India, and testing data from Indonesia. Based on the results of clustering using the two best models, both training data and testing data, more than half of data enter one cluster and the rest are distributed in the other two clusters. It can be concluded that the SARS-CoV-2 virus found in China, Japan, South Korea, Singapore, India, and Indonesia, mostly has similarities in virus sequences. The future research plan that can be done is to detect mutations that occur in each cluster using alignment and machine learning approaches.

ACKNOWLEDGEMENTS

The authors would like to thank the School of Electrical and Information Engineering, Bandung Institute of Technology for supporting this research, and Telkom University for the financial support.

REFERENCES

[1] K. Dhama *et al.*, “Coronavirus disease 2019–COVID-19,” *Clin. Microbiol. Rev.*, vol. 33, no. 4, pp. 1–48, 2020.

[2] World Health Organization, “Brazil: WHO Coronavirus Disease (COVID-19) Dashboard | WHO Coronavirus Disease (COVID-19) Dashboard,” *Who*, 2020. [Online]. Available: https://covid19.who.int/. [Accessed: 05-Mar-2021].

[3] World Health Organization, “Covid-19 in Indonesia,” 2020. [Online]. Available: https://covid19.who.int/region/searo/country/id. [Accessed: 01-Dec-2020].

[4] X.-W. Xu *et al.*, “Clinical findings in a group of patients infected with the 2019 novel coronavirus (SARS-Cov-2) outside of Wuhan, China: retrospective case series,” *BMJ*, p. m606, Feb. 2020.

[5] L. Zhang, F. Shen, F. Chen, and Z. Lin, “Origin and Evolution of the 2019 Novel Coronavirus,” *Clin. Infect. Dis.*, vol. 71, no. 15, pp. 882-883, Jul. 2020.

[6] M. I. Khan *et al.*, “Comparative genome analysis of novel coronavirus (SARS-CoV-2) from different geographical locations and the effect of mutations on major target proteins: An in silico insight,” *PLoS One*, vol. 15, no. 9, p. e0238344, Sep. 2020.

[7] J.-M. Kim *et al.*, “Identification of Coronavirus Isolated from a Patient in Korea with COVID-19,” *Osong Public Heal. Res. Perspect.*, vol. 11, no. 1, pp. 3-7, Feb. 2020, doi: 10.24171/j.phrp.2020.11.1.02.

[8] V. M. Corman *et al.*, “Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR,” *Eurosurveillance*, vol. 25, no. 3, pp. 2000045, Jan. 2020, doi: 10.2807/1560-7917.ES.2020.25.3.2000045.

[9] J. F.-W. Chan *et al.*, “Genomic characterization of the 2019 novel human-pathogenic coronavirus isolated from a patient with atypical pneumonia after visiting Wuhan,” *Emerg. Microbes Infect.*, vol. 9, no. 1, pp. 221-236, Jan. 2020, doi: 10.1080/22221751.2020.1719902.

[10] R. Lu *et al.*, “Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding,” *Lancet*, vol. 395, no. 10224, pp. 565-574, Feb. 2020, doi: 10.1016/S0140-6736(20)30251-8.

[11] J. Wang, H. He, and D. V. Prokhorov, “A folded neural network autoencoder for dimensionality reduction,” *Procedia Comput. Sci.*, vol. 13, pp. 120-127, 2012, doi: 10.1016/j.procs.2012.09.120

[12] A. Maćkiewicz and W. Ratajczak, “Principal components analysis (PCA),” *Comput. Geosci.*, vol. 19, no. 3, pp. 303-342, Mar. 1993, doi: 10.1016/0098-3004(93)90090-R.

[13] Adiwijaya, U. N. Wisesty, E. Lisnawati, A. Aditsania, and D. S. Kusumo, “Dimensionality Reduction using Principal Component Analysis for Cancer Detection based on Microarray Data Classification,” *J. Comput. Sci.*, vol. 14, no. 11, pp. 1521-1530, Nov. 2018, doi: 10.3844/jcssp.2018.1521.1530.

[14] T. Konishi, S. Matsukuma, H. Fuji, D. Nakamura, N. Satou, and K. Okano, “Principal Component Analysis applied directly to Sequence Matrix,” *Sci. Rep.*, vol. 9, no. 1, p. 19297, Dec. 2019.

[15] Y. Li, X. Luo, M. Chen, Y. Zhu, and Y. Gao, “An Autoencoder-Based Dimensionality Reduction Algorithm for Intelligent Clustering of Mineral Deposit Data,” 2020, pp. 408-415.

[16] M. Leyli-Abadi, L. Labiod, and M. Nadif, “Denoising Autoencoder as an Effective Dimensionality Reduction and Clustering of Text Data,” Advances in Knowledge Discovery and Data Mining, pp. 801-813, 2017.

[17] M. Alkhayrat, M. Aljnidi, and K. Aljoumaa, “A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA,” *J. Big Data*, vol. 7, no. 1, p. 9, Dec. 2020.

[18] M. Sirshar, S. Saleem, M. U. Ilyas, M. M. Khan, M. S. Alkatheiri, and J. S. Alowibdi, “Big Data Dimensionality Reduction for Wireless Sensor Networks Using Stacked Autoencoders,” Research & Innovation Forum, 2019, pp. 391-400.

[19] GISAID, “EpiCov TM,” 2020. [Online]. Available: https://www.epicov.org/epi3/frontend#1219a7. [Accessed: 21-Nov-2020].

[20] U. N. Wisesty, T. R. Mengko, and A. Purwarianti, “Gene mutation detection for breast cancer disease: A review,”

IOP Conf. Ser. Mater. Sci. Eng., vol. 830, p. 032051, May 2020, doi: 10.1088/1757-899X/830/3/032051.

[21] G. Mendizabal-Ruiz, I. Román-Godínez, Sulema Torres-Ramos, Ricardo A. Salido-Ruiz, J. A. Morales, "On DNA numerical representations for genomic similarity computation," *PLoS One*, no. ii, pp. 1-27, 2017, doi: 10.1371/journal.pone.0173288.

[22] S. S. Mohamed Ali, A. H. Alsaeedi, D. Al-Shammary, H. H. Alsaeedi, and H. W. Abid, "Efficient intelligent system for diagnosis pneumonia (SARS-COVID19) in X-Ray images empowered with initial clustering," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, p. 241-251, Apr. 2021.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[24] Y. Wang, H. Yao, S. Zhao, and Y. Zheng, "Dimensionality reduction strategy based on auto-encoder," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service-ICIMCS '15*, 2015, pp. 1-4, doi: 10.1145/2808492.2808555.

[25] D. P. Kingma and J. Ba, "Adam: a Method for Stochastic Optimization," *3rd Int. Conf. Learn. Represent.*, 2015.

[26] Y. Sun, "The Neural Network of One-Dimensional Convolution-An Example of the Diagnosis of Diabetic Retinopathy," *IEEE Access*, vol. 7, pp. 69657–69666, 2019, doi: 10.1109/ACCESS.2019.2916922.

[27] M. Capó, A. Pérez, and J. A. Lozano, "An efficient K-means clustering algorithm for tall data," *Data Min. Knowl. Discov.*, vol. 34, no. 3, pp. 776-811, May 2020.

[28] Y. Li and H. Wu, "A Clustering Method Based on K-Means Algorithm," *Phys. Procedia*, vol. 25, pp. 1104-1109, 2012, doi: 10.1016/j.phpro.2012.03.206.

[29] S. Sarkar, V. Melnykov, and R. Zheng, "Gaussian mixture modeling and model-based clustering under measurement inconsistency," *Adv. Data Anal. Classif.*, vol. 14, no. 2, pp. 379-413, Jun. 2020.

[30] Y. G. Jung, M. S. Kang, and J. Heo, "Clustering performance comparison using K -means and expectation maximization algorithms," *Biotechnol. Biotechnol. Equip.*, vol. 28, no. sup1, pp. S44-S48, Nov. 2014, doi: 10.1080/13102818.2014.949045.

[31] S. Zhou, Z. Xu, and F. Liu, "Method for Determining the Optimal Number of Clusters Based on Agglomerative Hierarchical Clustering," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 12, pp. 3007–3017, Dec. 2017, doi: 10.1109/TNNLS.2016.2608001.

[32] L. AbdAllah and I. Shimshoni, "Mean Shift Clustering Algorithm for Data with Missing Values," Data Warehousing and Knowledge Discovery, pp. 426-438. 2014.

[33] Nidhi and K. A. Patel, "An Efficient and Scalable Density-based Clustering Algorithm for Normalize Data," *Procedia Comput. Sci.*, vol. 92, pp. 136-141, 2016, doi: 10.1016/j.procs.2016.07.336.

[34] A. Starczewski and A. Krzyzak, "Performance evaluation of the silhouette index," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 9120, pp. 49-58, 2015.

[35] D. Ongkadinata and F. P. Putri, "Quality and size assessment of quantized images using K-Means++ clustering," *Bull. Electr. Eng. Informatics*, vol. 9, no. 3, pp. 1183-1188, Jun. 2020, doi: 10.11591/eei.v9i3.1985.

BIOGRAPHIES OF AUTHORS

Untari Novia Wisesty received the Bachelor and Master degree on Informatics Engineering from Telkom Institute of Technology (now Telkom University), Bandung, Indonesia in 2010 and 2012. She is now a Doctoral student in School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia. Since 2010, she joined Telkom University as a lecturer in School of Computing. Her research interest includes machine learning, deep learning, and bioinformatics. Scopus ID: 55597439000, Researcher ID: AAD-5807-2021, Publon ID: 4214913, Orcid: https://orcid.org/0000-0001-5803-9643.

Tati Rajab Mengko received the Ir, BS+ on Electrical Engineering from Institut Teknologi Bandung, Bandung, Indonesia in 1977, and Dr. Eng from ENSERG- INPG- Grenoble France in 1985. Since 1978, she joined School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia, and she received Professor in image processing of School of Electrical Engineering and Informatics, Bandung Institute of Technology in 2006. She is now head of biomedical engineering research division. Her research interest includes Image Processing and Instrumentation in Biomedical Engineering. Scopus ID: 14019972600.