7st International Seminar of Research Month 2022
Volume 2023
http://dx.doi.org/10.11594/nstp.2023.3338

NST Proceedings
OPEN ACCESS

Conference Paper

Classification of Covid-19 RT-PCR Test Results Using Auto-encoder And Random Forest

Andreas Nugroho Sihananto[1]*, Eristya Maya Safitri[2], Arif Widiasan Subagio[1], Muhammad Dafa Ardiansyah[1], Aditya Primayudha[1]

[1]Department of Informatics, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, 60294, Indonesia
[2]Department of Information System, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, 60294, Indonesia

Corresponding author:
E-mail:
andreas.nugroho.jarkom@upnja-tim.ac.id

ABSTRACT

Corona Virus Disease (COVID-19) is a new type of virus that emerged at the end of 2019. COVID-19 has become a pandemic due to the increase in the number of cases taking place very quickly and has spread to all corners of the world. The World Health Organization (WHO) recommends the use of the Reverse Transcription-Polymerase Chain Reaction (RT-PCR) method as a way to test the diagnosis of COVID-19 infection. This study builds a classification system for the COVID-19 RT-PCR test results by applying the *Auto-encoder* algorithm and the Random Forest classification. The dataset used is the result of the RT-PCR test from one of the hospitals in Brazil. The method used is the *Auto-encoder* to process the dataset features first and the Random Forest algorithm to classify the RT-PCR test results that have positive and negative labels. From this process, it can be seen that the *Auto-encoder* model can process datasets well and the classification carried out using Random Forest can classify with an accuracy of 87.2%.

Keywords: COVID-19, RT-PCR, classification, auto encoder, random forest

Introduction

In December 2019, a new virus emerged in the city of Wuhan, China, known as the coronavirus. At first, the virus was thought to be caused by exposure to a food market that sells many species of live animals. From December 18 to December 29, 2019, 5 patients were treated with Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2). The emergence of this corona virus has attracted the attention of the world, finally on January 30, 2020, the World Health Organization (WHO) declared that Coronavirus Disease 2019 (COVID-19) is a public health pandemic emergency that is of international concern. The increase in the number of COVID-19 cases took place very quickly and spread to all corners of the world quickly (Pristiyono et al., 2021).

COVID-19 is a new type of disease or virus that has never been previously identified in humans. COVID-19 is transmitted from human to human through sneezing or coughing droplets (droplets) with a high infection rate, making it easy to spread. Common symptoms of people infected with COVID-19 are symptoms of respiratory disorders pneumonia and a high risk of death(Koh, 2020; Wang et al., 2020).

As COVID-19 has become a worldwide pandemic, WHO is looking for ways to diagnose whether a person has COVID-19 or not. Finally, WHO recommends the Reverse Transcription-Polymerase Chain Reaction (RT-PCR) method as the standard for diagnosing COVID-19 infection. The RT-PCR method functions to detect the presence of the virus in the patient's body through a

How to cite:
Sihananto, A. N., Safitri, E. M., Subagio, A. W., et al. (2023). Classification of covid-19 RT-PCR test results using auto-encoder and random forest. *7st International Seminar of Research Month 2022*. NST Proceedings. pages 237-243. doi: 10.11594/ nstp.2023.3338

polymerase chain reaction with a primer that specifically targets the SARS-CoV-2 genome so that the number of SARS-CoV-2 cDNA in the patient can be calculated(Agustina & Fajrunni'mah, 2020). Currently, the RT-PCR test is the most accurate method for diagnosing COVID-19 in humans.

With the RT-PCR test to diagnose COVID-19 which has been used by almost all health institutions in the world, it is possible to create a classification system for RT-PCR test results to classify people diagnosed as positive and people diagnosed negative for COVID-19. There are many algorithms for classifying big data like this, such as the K-Nearest Neighbors algorithm, Artificial Neural Network, Support Vector Machine, Random Forest, and many more. Each existing algorithm has different methods and results from one another(Betechuoh et al., 2006; Sabu & Sreekumar, 2017; Zhang et al., 2020).

The data used for classification can also be normalized first. Normalization is the process of scaling the attribute values of the data so that they can lie within a certain range(Petti et al., 2020). The data normalization that is often used is Min-Max Normalization, Z-Score Normalization, and Decimal Scaling Normalization. However, there is one Neutral Network algorithm that can be used to normalize data, namely the *Auto-encoder* Network algorithm (Betechuoh et al., 2006).

The purpose of this study is to build a classification system for the COVID-19 RT-PCR test results by applying the *Auto-encoder* algorithm and the Random Forest classification. This study uses a fairly large amount of data and various features so it is expected to produce a good level of classification accuracy.

Material and Methods

Preparation of termite nest sample

Dataset

In this study, we use datasets that are appropriate to the topic we are researching. The dataset used is named "COVID-19 Einstein 25 Full" which we got on Kaggle. This database contains data from patients at the Hospital Israelita Albert Einstein in Sao Paulo, Brazil. The data taken are the results of the SARS-CoV-2 RT-PCR test and blood data from the patients. The dataset consists of 24 features with 1 label indicating whether the patient is positive or not, the total data available is 14,681 data.

Auto-encoder

Auto-encoder is an artificial neural network whose goal is to copy input into output. *Auto-encoder* is an advanced concept of *artificial neural network* (ANN) which is used to study the data representation used as a dataset dimensionality reduction. Before the learning process, the auto-encoder was used as a dataset dimension reducer with feature extraction. *Auto-encoder* is a representation of ANN which is trained to copy input to output. This algorithm works by encoding the input data, which is then reconstructed as output data based on the previously encoded data. *Auto-encoders* consist of two parts which are *encoder* and *decoder*. The *encoder* is the part that encodes the input. While the *decoder* is the part that reconstructs the input data as output data based on the data that has been encoded(Deng et al., 2021).

Auto-encoder can be used in many ways. One of them is data compression. The stage of the auto-encoder in use as data compression begins with encoding the data using a predetermined weight and bias. Then the auto-encoder reconstructs the input data from the encoded data. After that, the loss function is calculated to know the performance of the encoding that has been done. This training process is carried out continuously until it reaches an acceptable value.

Random Forest

Random Forest (RF) is one of the well-known and popular classification algorithms. Due to its high level of accuracy and maturity, *Random Forest* has been implemented in many research focused on machine learning, including many in bioinformatics and medical imaging. RF consists of a collection of decision trees, each of which is a decision tree generated from the bagging algorithm, which forms a 'forest' classifier to determine a class. To train RF, it takes *two parameters*, namely the *number of trees in the forest* and the *number of randomly selected feature numbers* that are used to evaluate each node in the tree. The calculation of the accuracy level embedded in the random forest algorithm is called *Out of Bag Error* (OOB), which calculates the average value of the misclassification ratio of samples that are not used in the random forest training (Belgiu & Dragut, 2012).

Flow

Our research is divided into several stages. The flow of our research can be seen in Figure 1.

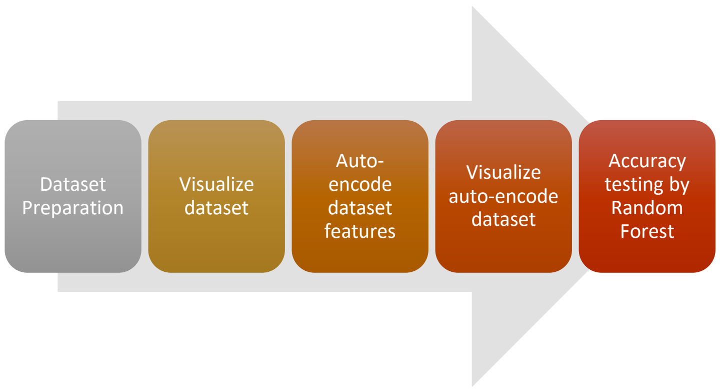
Dataset Preparation
Visualize dataset
Auto-encode dataset features
Visualize auto-encode dataset
Accuracy testing by Random Forest

Figure 1. Flow of research

a. *Dataset Preparation*
   In this stage, we prepare the dataset that we use and import the dataset into the program we created. We remove features that are not used in the classification process such as the *Date feature* and the *ID feature*. In addition, we changed the value of the Sex feature from initially categorical to numeric.
b. Visualization *Dataset*
   At this stage, we visualize the dataset before the auto-encoder in the form of a TSNE plot. This is done so that the overall distribution of the dataset between patients with positive and negative labels can be seen clearly.
c. *Auto-encode* Fitur pada *Dataset*
d. Before *auto-encoding* the dataset, we build the *auto-encoder* model first. The model we use can be seen in Figure 2.

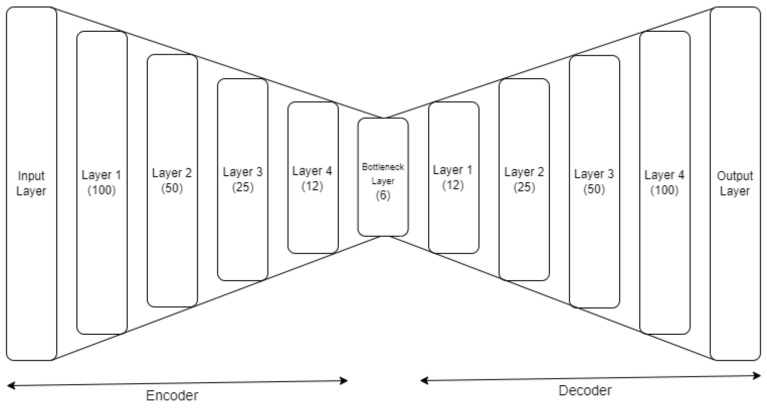

Figure 2. Model *Auto-encoder*

As seen in Figure 2, the model we created uses several *layers*. The *initial layer* is the input layer. Data input from the dataset enters the input layer which is then continued to the next layer. The data continues to be encoded until it reaches the *Bottleneck Layer* (8th layer). This layer contains the encoding results from the previous layers. After that, from the *bottleneck layer*, it will be continued by reconstructing the data that has been encoded before until it reaches the *output layer*. The *output layer* contains the results of the reconstruction of the previous layers.

e. Visualization *Dataset* After *Auto-encode*
At this stage, we visualize our data that has gone through the *auto-encoder* algorithm in the form of a *t-SNE* plot, the same as before. This is done to see the differences in the distribution of data between the initial dataset and the processed data.

f. Accuracy Testing using *Random Forest*
The encoded data will be split between the *train data* and the *test data*. This is intended to calculate the accuracy level of data compression which has been carried out using the *random forest classification* method. The data is split into 4 parts, namely *x train encoded, x test encoded, y train encoded,* and *y test encoded*. After that, the data is implemented into the *random forest algorithm* to check the level of accuracy.

Results and Discussion

In this result and discussion, we have several important meeting points that we can further analyze the results, such as the results of data visualization that has not been done by *auto-encoder*, results of data visualization that has been done by *auto-encoder*, training loss function graphs, and validation loss function graphs on *auto-encoder* process, as well as comparing the accuracy generated from data that has been *auto-encoder*ed with data that has not been *auto-encoder*ed with the random forest algorithm.

Data Visualization

Based on the above method and our purpose in conducting this research, we will compare the data that has been processed by *auto-encoder* with data that has not been processed by *auto-encoder*. In terms of the results of data visualization using the t-SNE plot, the following results are obtained:

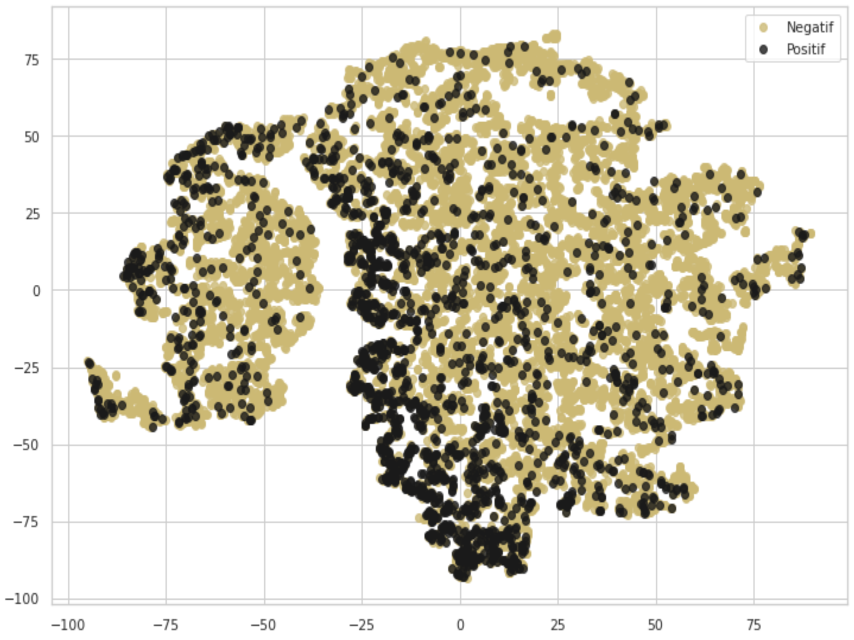

Figure 3. Data visualization without *Auto-encoder*

Visualization without Auto-encoder

In Figure 3, it can be seen that in the data visualization without an auto-encoder, the distribution of data through the t-SNE plot approach is not evenly distributed, and the label between negative and positive is still more one-sided.

Visualization with Auto-encoder

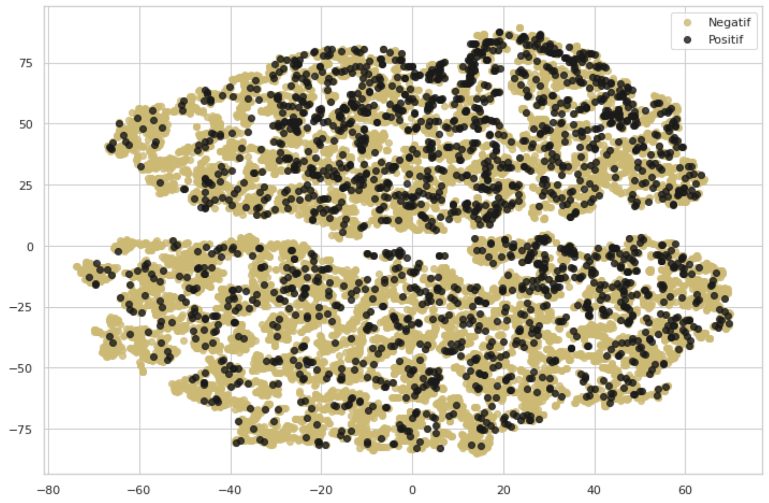

Figure 4. Data visualization with implementation *Auto-encoder*

While in Figure 4 it can be seen that in the data visualization using an auto-encoder, the distribution of data through the t-SNE plot approach has spread evenly and the distribution is almost equal, and the labels between negative and positive are already being divided equally.

Loss-function Graphic

The results of the training in the auto-encoder process are carried out repeatedly or by changing *hyperparameters* such as *epochs* (iterations) 20 times, where the data for each iteration results in significant changes between the intervals of the experiment repetition, such as the results of *loss (training loss function)* and *val_loss(*validation loss function*), batch_size*(number of sample sets used in each iteration) of 16, and hyperparameter shuffle which is useful for randomizing the contents of the dataset. The following Figure 5 show the graph of the training loss function and the validation loss function.

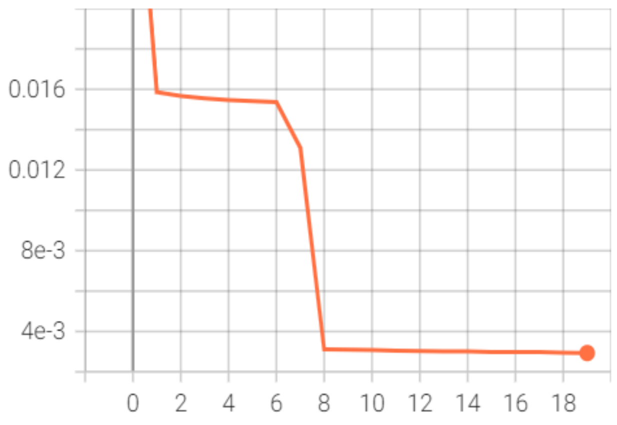

Figure 5. Training Loss Function Graphics

In Figure 5 it can be seen that the training loss function graph with the x-axis is the number of iterations that occur and the y-axis is the training loss value for each iteration, the graph indicates that in iterations 0 to 8 it is not too significant to increase the value of the initial training loss performance. from 0.307 to 0.131, while in iterations 9 to 10 it has a very significant performance increase so that in the 11th iteration the value of training loss drops drastically and remains the same until the 20th iteration, which is 0.0029 or 2,9x10^{-3}.

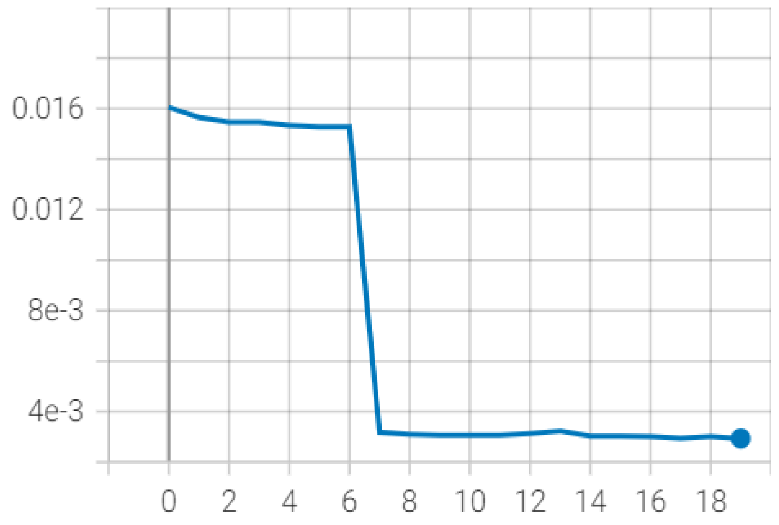

Figure 6. Validation loss function graphics

In Figure 6 it can be seen that the validation loss function graph is almost the same as the graph in Figure 5, namely the training loss function graph, that in iterations 0 to 7 it is not too significant to increase the value of the validation loss performance which was originally from 0.161 to 0.153, while in iterations 8 to 9 it has a very significant increase in performance so that in the 10th iteration the value training loss has decreased drastically (increased performance) and remains the same until the 20th iteration, which is 0.0026 atau 2,6x10^{-3}.

Meanwhile, there is also accuracy testing. Accuracy in machine learning models is a measurement used to determine which model is best at identifying relationships and patterns between variables in a data set based on input data, or *training data*. In this study, the data that has been processed by *auto-encoder* or which will not be classified using the Random Forest algorithm will be used. After modeling the training data that changes several hyperparameters such as *n_estimators* (the number of trees to be used in the random forest algorithm) as much as 1800 and *random_state* on *seed* 42, the result is an accuracy of each model. We use 2 different models of input data, which have the following different results as seen in Table 1.

Table 1. Accuracy comparison

Data using *Auto-encoder* (%)	**Data without *Auto-encoder* (%)**
89,70238338440586	86,02114402451481

The accuracy of the random forest algorithm model using data input with an auto-encoder is higher than the accuracy of the random forest algorithm model that uses data input without an auto-encoder.

Conclusion

Based on the results of the discussion that has been described, the following conclusions can be drawn:

1. It can be concluded that the data set with the *auto-encoder* applied is more evenly distributed than the data without the *auto-encoder*, as can be seen in the comparison of data visualization with t-SNE plotting.
2. The data that is being trained in the *auto-encoder* process does not require many iterations, because the more iterations the more results will not be too significant changes.
3. The accuracy of the model using the Random Forest algorithm that uses data input with an *auto-encoder* is higher in accuracy (89.7%) than data that uses data input without an *auto-encoder* (87.2%) which uses the same algorithm in the modeling.

Acknowledgment

This work was financially supported by LPPM Universitas Pembangunan Nasional "Veteran" Jawa Timur through –the Uber Publikasi Bereputasi 2022 Scheme.

References

Agustina, A. S., & Fajrunni'mah, R. (2020). Perbandingan metode RT-PCR dan tes rapid antibodi untuk deteksi COVID-19. *Jurnal Kesehatan Manarang*, *6*, 47–54.

Belgiu, M., & Dragut, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 24–31. Doi:10.1016/j.isprsjprs.2016.01.011

Betechuoh, B. L., Marwala, T., & Tettey, T. (2006). Autoencoder networks for HIV classification. *Current Science*, 1467–1473.

Deng, Y. C., Tang, X. H., Zhou, Z. Y., Yang, Y., & Niu, F. (2021). Application of machine learning algorithms in wind power: a review. *Energy Sources, Part A: Recovery, Utilization and Environmental Effects*, *00*(00), 1–22. https://doi.org/10.1080/15567036.2020.1869867

Koh, D. (2020). Occupational risks for COVID-19 infection. *Occupational Medicine*, *70*(1), 3–5. https://doi.org/10.1093/occmed/kqaa036

Petti, U., Baker, S., & Korhonen, A. (2020). A systematic literature review of automatic Alzheimer's disease detection from speech and language. *Journal of the American Medical Informatics Association*, *27*(11), 1784–1797. https://doi.org/10.1093/jamia/ocaa174

Pristiyono, Ritonga, M., Ihsan, M. A. Al, Anjar, A., & Rambe, F. H. (2021). Sentiment analysis of COVID-19 vaccine in Indonesia using Naïve Bayes Algorithm. *IOP Conference Series: Materials Science and Engineering*, *1088*(1), 012045. https://doi.org/10.1088/1757-899x/1088/1/012045

Sabu, A., & Sreekumar, K. (2017). Literature review of image features and classifiers used in leaf based plant recognition through image analysis approach. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*, (Icicct), 145–149. https://doi.org/10.1109/ICICCT.2017.7975176

Wang, C., Cheng, Z., Yue, X.-G., & McAleer, M. (2020). Risk Management of COVID-19 by Universities in China. *Journal of Risk and Financial Management*, *13*(2), 36. https://doi.org/10.3390/jrfm13020036

Zhang, H., Jiang, L., & Yu, L. (2020). Class-specific attribute value weighting for Naive Bayes. *Information Sciences*, *508*, 260–274. https://doi.org/10.1016/j.ins.2019.08.071