JIKO (Jurnal Informatika dan Komputer) Accredited KEMENDIKBUD RISTEK, No.105/E/KPT/2022
Vol. 6, No. 2, Agustus 2023, hlm. 138-143 p-ISSN: 2614-8897
DOI: 10.33387/jiko.v6i2.6389 e-ISSN: 2656-1948

# HYPERPARAMETER TUNING ON RANDOM FOREST FOR DIAGNOSE COVID-19

**Anna Baita[1*], Inggar Adi Prasetyo[2], Nuri Cahyono[3]**

[1,2,3]Faculty of Computer Science, Universitas Amikom Yogyakarta
[*]Email: [1]anna@amikom.ac.id, [2]inggar.25@students.amikom.ac.id , [3]nuricahyono@amikom.ac.id

(Received: 21 July 2023, Revised: 31 July 2023, Accepted: 4 August 2023)

**Abstract**

Diagnosis of Covid using the RT-PCR (Reverse Transcription Polymerase Chain Reaction) test requires high costs and takes a long time. For this reason, another method is needed that can be used to diagnose Covid-19 quickly and accurately. Random Forest is one of the popular classification algorithms for making predictive models. Random forest involves many hyperparameters that control the structure of each tree, the forest, and its randomness. Random Forest is a method which very sensitive to hyperparameter values, as their prediction accuracy can increase significantly when optimized hyperparameters are predefined and then adjusted according to the procedure. The purpose of doing hyperparameter tuning on the random forest algorithm is to increase accuracy in the diagnosis of covid-19. Searching for optimal values of hyperparameters is done by the Grid Search method and Random Search. The result explains that the Random Forest can be used to diagnose Covid-19 with an accuracy of 94%, and with hyperparameter tuning, it can increase the accuracy of the random forest by 2%.

**Keywords**: *Hyperparameter Tuning, Random Forest, Covid-19, Accuracy, Classification*



**Corresponding Author: Anna Baita*

## 1. INTRODUCTION (UPPERCASE, 10pt, bold)

Corona Virus Disease 2019 (Covid 19) is a global pandemic that has hit various countries. Even though the pandemic has been going on for more than 2 years, there are still 794,000 new cases of Covid-19 and more than 4,800 deaths reported in the last 28 days (12 June to 9 July 2023) [1]. Diagnosis of Covid-19 disease is generally carried out using the RT-PCR (Reverse Transcription Polymerase Chain Reaction) test [2]. The RT-PCR test can diagnose Covid disease fairly accurately [3]. However, the RT-PCR test requires quite a high cost [4] and is time-consuming [5]. Therefore, many studies have been developed to diagnose Covid disease quickly and accurately.

Machine learning is widely used in the health sector, including diagnosing covid[6]. Random Forest is a machine learning algorithm widely used to research Covid-19. Gupta's research[7] discusses predicting confirmed cases, deaths, and recoveries from Covid-19 using the Random Forest algorithm. Predictions are made based on the observation date, time, and region in India. According to this study, the random forest algorithm has better accuracy results when compared to the Decision Tree, Multinomial Logistic Regression, Neural network, and SVM algorithms. Rostami's research [8] discusses feature selection using random forests to diagnose Covid-19. The diagnosis of COVID-19 is made based on the results of routine blood tests. Aser's research [9] conducted a classification using the KNN and Random Forest algorithms to predict Covid 19. This research was conducted to predict the number of daily cases. Several deep learning techniques have also been implemented to diagnose Covid-19 based on chest X-ray images and lungs [10]–[12].

In Gupta and Aser's study, predictions of COVID-19 were not used for diagnosis but to predict the number of cases. Previous research which use images and test blood approaches to diagnose Covid-19 requires expensive fees, specialist staff, and a certified laboratory. Therefore, in this study, we proposed diagnosing Covid using symptoms and parameters everyone can check independently.

Random forest involves many hyperparameters that control the structure of each tree, the forest, and its randomness[13].

Random Forest is a method which very sensitive to hyperparameter values, as their prediction accuracy can increase significantly when optimized hyperparameters are predefined and then adjusted according to the procedure[14].

Imbalanced class allocation affects training classification process, leading to an unfavorable bias for majority class [15]. Smote is a method commonly used to overcome distribution inequality in imbalanced data [16], [17].

Therefore we will perform hyperparameter tuning on the random forest classifier to improve the accuracy of the random forest classifier in this Covid-19 diagnosis.

## 2. RESEARCH METHOD

The research flow is illustrated in Figure 1 below:

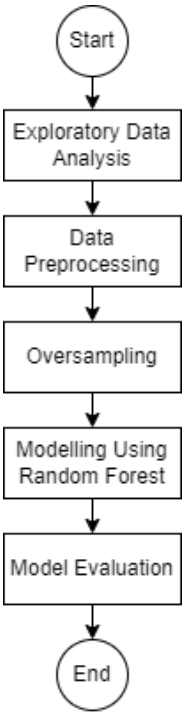


Figure 1. Research Flow

### 2.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) examines available datasets to find patterns, find anomalies, test hypotheses and check assumptions using statistical measures [18]. EDA is used to determine the important characteristics of the dataset used.

### 2.2 Data Preprocessing

Data preprocessing is the process of preparing data for analysis. The dataset is preprocessed to check missing values before executing it to the algorithm. Data transformation ensures the data's compatibility with the algorithm used in the mining process. Feature selection is selecting features to be used in model building. In this research, the feature selection process is carried out by looking for the value of Pearson's Correlation. Pearson's Correlation is the most popular method for measuring the direction and strength of a linear relationship between two variables [19].

### 2.3 Oversampling

Synthetic Minority Oversampling Technique (SMOTE) [20] is one of the oversampling methods in dealing with imbalanced datasets. SMOTE generates a new sample from the minority class without repetition. SMOTE generates synthetic minority examples using the k nearest neighbors of the allowed minority examples, as the following figure 2 shows

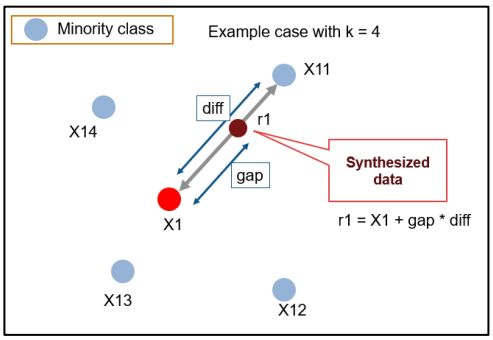


Figure 2. The principle of the synthetic minority oversampling technique (SMOTE) [21]

SMOTE can improve accuracy results compared to classifications without balancing the dataset [22].

### 2.4 Modelling Using Random Forest

#### 2.4.1 Random Forest

A Random Forest model contains many decision trees[23]. A Random Forest is an effective tool in prediction. Random forest was first introduced by Breinman [24]. Determination of the classification with Random Forest is done based on the results of voting from the formed decision tree. The way Random Forest works is shown in the following figure 3:

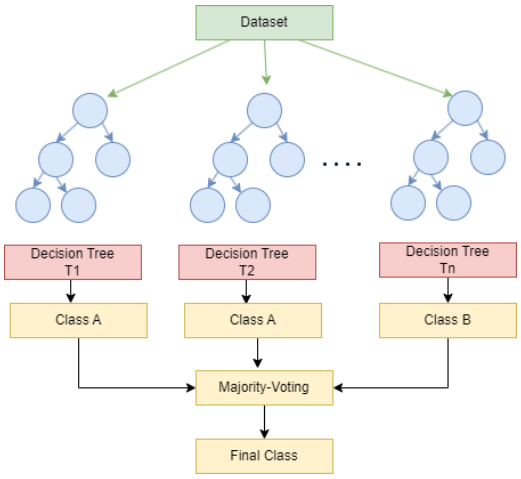


Figure 3. Random Forest Classifier

### 2.4.1 Tuning Random Forest

The Random Forest algorithm has hyperparameters that can affect its performance [13]. Tuning is finding the optimal hyperparameters of an algorithm during the learning process. Hypermeter tuning is done by training and evaluating the model using different hyperparameter combinations using k-fold validation and selecting a set of hyperparameters that produce low error values [25]. There are 2 methods commonly used in hyperparameter tuning, namely grid search and random search [26]. Grid search is a method in hyperparameter tuning by trying all combinations of parameters in the search space. The combination of parameters that is want to try is stored in the grid. Grid Search will look for the parameter combination that has the smallest error [27]. Random Search exploration parameters are taken randomly from a combination of parameters in the search space [28].

### 2.5 Model Evaluation

A confusion matrix is used to evaluate the model that has been built. *True positives* are the number of predicted values worth true that identify those affected by Covid 19. *True negatives* are the predicted values that are true for data that are not affected by Covid 19. *False positives* are predictions that are false for data affected by Covid 19. Furthermore, *False negatives* are predictions with a false value for data unaffected by Covid 19. The analysis is carried out with accuracy, precision, recall, f1-score, and ROC-AUC value metrics.

## 3. Result And Discussion

The data used in this study comes from Kaggle[29]. In the Exploratory data analysis process, information was obtained that the dataset consisted of 5,434 data. The dataset has 20 features, namely: *Breathing Problems, Fever, Dry Cough, Sore throat, Running Nose, Asthma, Chronic Lung Disease, Headache, Heart Disease, Diabetes, HyperTension, Fatigue, Gastrointestinal, Abroad travel, Contact with COVID Patient, Attended Large Gathering, Visited Public Exposed Places, Family working in Public Exposed Places, Wearing Masks, Sanitization from Market*. All features of the dataset are of type object (data categorical). The target label/variable from the dataset is Covid-19 which has the values 'Yes' and 'No'. The covid-19 dataset is imbalanced dataset as shown in Figure 4 below:

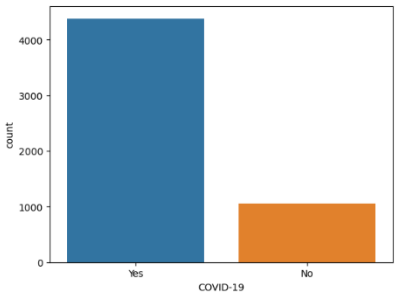


Figure 4. Target Classess

Class 'No' consists of 1051 data, while class 'Yes' consists of 4383 data.

### 3. 1 Data Preprocessing

#### 3.1.1 Data Cleaning

First, we check for empty data. The results of checking the empty data are shown in the following figure:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5434 entries, 0 to 5433
Data columns (total 21 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Breathing Problem                     5434 non-null   object
 1   Fever                                 5434 non-null   object
 2   Dry Cough                             5434 non-null   object
 3   Sore throat                           5434 non-null   object
 4   Running Nose                          5434 non-null   object
 5   Asthma                                5434 non-null   object
 6   Chronic Lung Disease                  5434 non-null   object
 7   Headache                              5434 non-null   object
 8   Heart Disease                         5434 non-null   object
 9   Diabetes                              5434 non-null   object
 10  Hyper Tension                         5434 non-null   object
 11  Fatigue                               5434 non-null   object
 12  Gastrointestinal                      5434 non-null   object
 13  Abroad travel                         5434 non-null   object
 14  Contact with COVID Patient            5434 non-null   object
 15  Attended Large Gathering              5434 non-null   object
 16  Visited Public Exposed Places         5434 non-null   object
 17  Family working in Public Exposed Places  5434 non-null   object
 18  Wearing Masks                         5434 non-null   object
 19  Sanitization from Market              5434 non-null   object
 20  COVID-19                              5434 non-null   object
dtypes: object(21)
memory usage: 891.6+ KB
```

Figure 5. Dataset Information

From Figure 5, it can be seen that the Covid-19 dataset does not have a missing value.

#### 3.1.2 Data Transformation

In this step, all variables from the dataset are converted into numeric types using encoder labels. The process and results of data transformation are shown in Figure 6 below:

```
In [10]: label_encoder = LabelEncoder()
         for col in df.columns:
             df[col] = label_encoder.fit_transform(df[col])

In [11]: num_cols = df.select_dtypes(include=np.number).columns.tolist()
         print("Numerical Variables:")
         print(num_cols)

         Numerical Variables:
         ['Breathing Problem', 'Fever', 'Dry Cough', 'Sore throat', 'Runnin
         g Nose', 'Asthma', 'Chronic Lung Disease', 'Headache', 'Heart Dise
         ase', 'Diabetes', 'Hyper Tension', 'Fatigue ', 'Gastrointestinal
         ', 'Abroad travel', 'Contact with COVID Patient', 'Attended Large
         Gathering', 'Visited Public Exposed Places', 'Family working in Pu
         blic Exposed Places', 'Wearing Masks', 'Sanitization from Market',
         'COVID-19']
```

Figure 6. Data Transformation

### 3.1.3 Feature Selection

Feature selection is done by looking at the correlation of each feature to the target variable. Features used in the classification process have a correlation value > 0.1. From the feature selection process, 10 features were obtained with a high correlation with the target variable. These features include *Respiratory Problems, Fever, Dry Cough, Sore Throat, Hypertension, Travel* Overseas, *Contact with COVID Patients, Attending Large Meetings, Visiting Public Exposed Places, and Families Work in Public Exposed Places.* The process and results of feature selection are shown in Figure 7 below:

```
In [16]: #Correlation with output variable
         cor_target = abs(cor['COVID-19'])
         #Selecting highly correlated features
         relevant_features = cor_target[cor_target>0.1]
         relevant_features

Out[16]: Breathing Problem                        0.443764
         Fever                                   0.352891
         Dry Cough                               0.464292
         Sore throat                             0.502848
         Hyper Tension                           0.102575
         Abroad travel                           0.443875
         Contact with COVID Patient              0.357122
         Attended Large Gathering                0.390145
         Visited Public Exposed Places           0.119755
         Family working in Public Exposed Places 0.160208
         COVID-19                                1.000000
         Name: COVID-19, dtype: float64
```

Figure 7. Feature Selection

### 3.1.4 Oversampling

```
from collections import Counter
from imblearn.over_sampling import BorderlineSMOTE
from imblearn.over_sampling import SMOTE
print('Original dataset shape %s' % Counter(y))

Original dataset shape Counter({1: 4383, 0: 1051})

sm = SMOTE(random_state=42)

X_res, y_res = sm.fit_resample(X, y)

print('Resampled dataset shape %s' % Counter(y_res))

Resampled dataset shape Counter({1: 4383, 0: 4383})
```

Figure 8. Oversampling

In Figure 8, the SMOTE method generates samples in class 0 (No), so there are 4383 data. Thus, classes 0 and 1 are now balanced.

## 3. 2 Modelling

After the data is oversampled, the data is used to build a model using the Random Forest algorithm. To improve the results of the accuracy of the hyperparameter tuning process is carried out. The list of search spaces used in hyperparameter tuning is shown in Table 1 below:

Table 1. Search Space

| No | Parameter | Value |
|---|---|---|
| 1 | N_estimator | [200,300,400,500] |
| 2 | Max_features | [‘auto’,’sqrt’,’log2’] |
| 3 | Max_depth | [4,5,6,7,8] |
| 4 | Min_samples_split | [2,3,4,5] |
| 5 | Min_samples_leaf | [1,2,3,4,5] |

RandomSearchCV and GridSearchCV carry out the optimal parameter search process. The Cross Validation (CV) used in this study is 10-fold. RandomSearch CV process is faster than GridSearchCV. GridSearch CV will build a model combining each value parameter in the search space. The total number of models built during the grid search process is 12,000 models. Therefore, the computational process of Gridsearch is very long. The optimal parameter results from the GridsearchCV and RandomSearchCV processes are shown in Table 2 below:

Table 2. Best Parameter

| No | Parameter | Grid Search | Random Search |
|---|---|---|---|
| 1 | N_estimator | 300 | 300 |
| 2 | Max_features | auto | auto |
| 3 | Max_depth | 8 | 8 |
| 4 | Min_samples_split | - | 3 |
| 5 | Min_samples_leaf | - | - |

N_estimator represents the number of trees in a random forest. Usually, more trees will be able to study the data pattern better. In Random Forest, a random subset of features is considered to find the best split. Max features help determine how many features should be included in the split. It can take four values: "*auto*", "*sqrt*", "*log2*", and "*none*".

*Max_depth* is one of the most important hyperparameters in improving model accuracy. Increasing the tree depth increases the model accuracy up to a certain limit, after which it gradually decreases due to model overfitting. Specifies the minimum number of samples an inner node must contain to split into other nodes. In this case if the value of *min_samples_splits* is very low, the tree will continue to grow, and overfitting will begin.

*Min_samples_leaf* specifies the minimum number of samples a node should contain after splitting. It also helps reduce overfitting when there are many parameters.

The modeling results obtained without parameter tuning and with hyperparameter tuning are shown in table 3 below:

Table 3. Experiment Result

| No | Metric (mean) | Without Hyper parameter Tuning | Grid Search | Random Search |
|---|---|---|---|---|
| 1 | Accuracy | 0.94 | 0.96 | 0.96 |
| 2 | Precision | 1 | 0.98 | 0.98 |
| 3 | Recall | 0.93 | 0.93 | 0.93 |
| 4 | F1-Score | 0.96 | 0.95 | 0.95 |
| 5 | ROC-AUC | 0.99 | 0.99 | 0.99 |

Hyperparameter tuning using grid search and random search can increase system accuracy by 2%. Hyperparameter tuning with grid search and random search has no difference in results. However, the computational process of grid search requires more time when compared to random search.

The default value of the random forest parameter using the Scikit learn library can produce an optimal model. Hyperparameter tuning is effective in increasing accuracy but not for other metrics.

## 4. CONCLUSION

Based on the above reviews, Hyperparameter tuning in random forest can increase the accuracy value but not necessarily improve Precision, Recall or F1-Score. Determining the value of the search space will affect the results of hyperparameter tuning.

## 5. REFERENCES


[1]. WHO, "Weekly epidemiological update on COVID-19 - 13 July 2023," *who.int*, Jul. 13, 2023. https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19---13-july-2023 (accessed Jul. 14, 2023).

[2]. M. Maniruzzaman *et al.*, "COVID-19 diagnostic methods in developing countries," *Environmental Science and Pollution Research*, vol. 29, no. 34. Springer Science and Business Media Deutschland GmbH, pp. 51384–51397, Jul. 01, 2022. doi: 10.1007/s11356-022-21041-z.

[3]. R. Pu *et al.*, "The screening value of RT-LAMP and RT-PCR in the diagnosis of COVID-19: systematic review and meta-analysis," *Journal of Virological Methods*, vol. 300. Elsevier B.V., Feb. 01, 2022. doi: 10.1016/j.jviromet.2021.114392.

[4]. A. Kadir, *SE-DIRJEN-YANKES-TTG-BATAS-TARIF-TERTINGGI-PEMERIKSAAN-RT-PCR*. Indonesia, 2021.

[5]. M. Döhla *et al.*, "Rapid point-of-care testing for SARS-CoV-2 in a community screening setting shows low sensitivity," *Public Health*, vol. 182, pp. 170–172, May 2020, doi: 10.1016/j.puhe.2020.04.009.

[6]. A. S. Kwekha-Rashid, H. N. Abduljabbar, and B. Alhayani, "Coronavirus disease (COVID-19) cases analysis using machine-learning applications," *Applied Nanoscience (Switzerland)*, vol. 13, no. 3, pp. 2013–2025, Mar. 2023, doi: 10.1007/s13204-021-01868-7.

[7]. V. K. Gupta, A. Gupta, D. Kumar, and A. Sardana, "Prediction of COVID-19 confirmed, death, and cured cases in India using random forest model," *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 116–123, Jun. 2021, doi: 10.26599/BDMA.2020.9020016.

[8]. M. Rostami and M. Oussalah, "A novel explainable COVID-19 diagnosis method by integration of feature selection with random forest," *Inform Med Unlocked*, vol. 30, Jan. 2022, doi: 10.1016/j.imu.2022.100941.

[9]. W. Aser, H. Samosir, and T. Gantini, "Analisis Dataset COVID-19 menggunakan Algoritma KNN dan Random Forest," *Jurnal Strategi*, vol. 4, May 2022.

[10]. Y. H. Bhosale and K. S. Patnaik, "Application of Deep Learning Techniques in Diagnosis of Covid-19 (Coronavirus): A Systematic Review," *Neural Processing Letters*. Springer, 2022. doi: 10.1007/s11063-022-11023-0.

[11]. R. Hertel and R. Benlamri, "A deep learning segmentation-classification pipeline for X-ray-based COVID-19 diagnosis," *Biomedical Engineering Advances*, vol. 3, p. 100041, Jun. 2022, doi: 10.1016/j.bea.2022.100041.

[12]. S. Aslani and J. Jacob, "Utilisation of deep learning for COVID-19 diagnosis," *Clinical Radiology*, vol. 78, no. 2. W.B. Saunders Ltd, pp. 150–157, Feb. 01, 2023. doi: 10.1016/j.crad.2022.11.006.

[13]. P. Probst, M. N. Wright, and A. L. Boulesteix, "Hyperparameters and tuning strategies for random forest," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 3. Wiley-Blackwell, May 01, 2019. doi: 10.1002/widm.1301.

[14]. M. Daviran, A. Maghsoudi, R. Ghezelbash, and B. Pradhan, "A new strategy for spatial predictive mapping of mineral prospectivity: Automated hyperparameter tuning of random forest approach," *Comput Geosci*, vol. 148, Mar. 2021, doi: 10.1016/j.cageo.2021.104688.

[15]. D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data," *IEEE Trans Neural Netw Learn Syst*, 2022, doi: 10.1109/TNNLS.2021.3136503.

[16]. Asniar, N. U. Maulidevi, and K. Surendro, "SMOTE-LOF for noise identification in imbalanced data classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3413–3423, Jun. 2022, doi: 10.1016/j.jksuci.2021.01.014.

[17]. A. Indrawati, "Penerapan Teknik Kombinasi Oversampling Dan Undersampling Untuk Mengatasi Permasalahan Imbalanced Dataset," *Jurnal Informatika dan Komputer)*, vol. 4, no. 1, 2021, doi: 10.33387/jiko.

[18]. S. K. M. Mukhiya and U. Ahmed, *Hands-on exploratory data analysis with Python : perform EDA techniques to understand, summarize, and investigate your data*, vol. 1. Birmingham-Mumbay: Packt Publishing, 2020.

[19]. C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Frontiers in Energy Research*, vol. 9. Frontiers Media S.A., Mar. 29, 2021. doi: 10.3389/fenrg.2021.652801.

[20]. M. Utari, "Implementation of Data Mining for Drop-Out Prediction using Random Forest Method," 2020.

[21]. S. Satpathy, "SMOTE for Imbalanced Classification with Python," Oct. 06, 2020. https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/ (accessed Jul. 18, 2023).

[22]. L. K. Xin and N. binti A. Rashid, "Prediction of depression among women using random oversampling and random forest," in *2021 International Conference of Women in Data Science at Taif University, WiDSTaif 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021. doi: 10.1109/WIDSTAIF52235.2021.9430215.

[23]. F. Khozeimeh *et al.*, "RF-CNN-F: random forest with convolutional neural network features for coronary artery disease diagnosis based on cardiac magnetic resonance," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-15374-5.

[24]. L. Breiman, "Random Forests," *Mach Learn*, vol. 45, pp. 5–32, 2001, doi: https://doi.org/10.1023/A:1010933404324.

[25]. D. Markovics and M. J. Mayer, "Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction," *Renewable and Sustainable Energy Reviews*, vol. 161, Jun. 2022, doi: 10.1016/j.rser.2022.112364.

[26]. L. Torre-Tojal, A. Bastarrika, A. Boyano, J. M. Lopez-Guede, and M. Graña, "Above-ground biomass estimation from LiDAR data using random forest algorithms," *J Comput Sci*, vol. 58, Feb. 2022, doi: 10.1016/j.jocs.2021.101517.

[27]. M. Fajri and A. Primajaya, "Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search," 2023. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[28]. R. Valarmathi and T. Sheela, "Heart disease prediction using hyper parameter optimization (HPO) tuning," *Biomed Signal Process Control*, vol. 70, Sep. 2021, doi: 10.1016/j.bspc.2021.103033.

[29]. H. Hariskrishnan, "Symptoms and COVID Presence (May 2020 data)," *Kaggle*, May 2020. https://www.kaggle.com/datasets/hemanthhari/symptoms-and-covid-presence (accessed May 21, 2023).