

# Hydrophobicity signal analysis for robust SARS-CoV-2 classification

Mohammad Jamhuri<sup>1,2</sup>, Mohammad Isa Irawan<sup>1</sup>, Imam Mukhlash<sup>1</sup>, Ni Nyoman Tri Puspaningsih<sup>3</sup>

<sup>1</sup>Department of Mathematics, Faculty of Science and Data Analytics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

<sup>2</sup>Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia

<sup>3</sup>Department of Chemistry, Faculty of Science and Technology, Universitas Airlangga, Surabaya, Indonesia

## Article Info

### Article history:

Received Jun 25, 2024

Revised Sep 24, 2024

Accepted Sep 30, 2024

### Keywords:

Convolutional neural networks

Genetic sequencing

Kyte-Doolittle scale

Machine learning

Virus identification

## ABSTRACT

Rapid and accurate classification of viral pathogens is critical for effective public health interventions. This study introduces a novel approach using convolutional neural networks (CNN) to classify SARS-CoV-2 and non-SARS-CoV-2 viruses via hydrophobicity signal derived from DNA sequences. Conventional machine learning methods grapple with the variability of viral genetic material, requiring fixed-length sequences and extensive preprocessing. The proposed method transforms genetic sequences into image-based representations, enabling CNNs to handle complexity and variability without these constraints. The dataset includes 8,143 DNA sequences from seven coronaviruses, translated into amino acid sequences and evaluated for hydrophobicity. Experimental results demonstrate that the CNN model achieves superior performance, with an accuracy of over 99.84% in the classification task. The model also performs well with extended sequence lengths, showcasing robustness and adaptability. Compared to previous studies, this method offers higher accuracy and computational efficiency, providing a reliable solution for rapid virus detection with potential applications in bioinformatics and clinical settings.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Mohammad Isa Irawan

Department of Mathematics, Faculty of Science and Data Analytics

Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia

Email: mii@its.ac.id

## 1. INTRODUCTION

Genomics has transformed the understanding of biological processes and disease mechanisms. High-throughput sequencing technologies have generated vast amounts of genomic data, necessitating advanced computational methods for analysis [1]. The rapid mutation and spread of viruses like SARS-CoV-2 underscore the urgency for robust genomic analysis techniques to track evolving pathogens [2].

Hydrophobicity, a fundamental property of amino acids, significantly impacts protein folding, stability, and interactions [3]. Hydrophobic interactions, where non-polar amino acid residues cluster away from water, help maintain protein stability and facilitate proper folding [4]. A detailed understanding of the hydrophobic profile of proteins is essential for revealing the mechanisms behind their structure and function [5].

Hydrophobic protein regions maintain structural integrity by being enclosed within the core, contributing to stability and three-dimensional conformation [6]. These regions are crucial for protein-protein and protein-ligand interactions, vital for understanding viral mechanisms and identifying therapeutic targets [7]. Machine learning models can leverage hydrophobicity profiles as biologically relevant features, effectively representing genetic information and facilitating data analysis through convolutional neural networks (CNNs) [8].

By converting genetic sequences into hydrophobicity-based visual representations, the complexity of viral genetic data can be managed more efficiently, surpassing the capabilities of traditional sequence-based methods [5].

Classifying SARS-CoV-2 and non-SARS-CoV-2 sequences is crucial for understanding viral genomes and has significant implications for drug discovery [9]. Accurate classification helps identify conserved regions within the SARS-CoV-2 genome, essential for developing antiviral drugs targeting key viral functions and replication processes [10]. This classification also aids in monitoring mutations that may affect the virus's susceptibility to treatments, guiding the development of effective drugs against emerging variants [11]. Furthermore, insights from this classification can inform the design of broad-spectrum antivirals by revealing common vulnerabilities across different viruses, which is crucial for preparing against future coronavirus outbreaks [12]. Additionally, accurate classification assists in identifying epitopes for vaccine development, accelerates drug discovery through high-throughput screening and structure-based design, and ultimately leads to more effective therapeutic interventions and improved public health outcomes [13].

Traditional methods, such as polymerase chain reaction (PCR) [3], sequencing [4], and serological assays [5], often grapple with the complexity and variability of viral genetic material, underscoring the need for advanced computational approaches. Early machine-learning techniques applied to viral studies have typically focused on the direct analysis of DNA or RNA sequences [14]. These techniques typically require fixed input lengths and involve extensive preprocessing to extract meaningful features from genetic data [15]. Conventional sequence-based machine learning models [16] face significant challenges in handling the variability and complexity of viral genetic material, limiting their effectiveness and adaptability. Studies on metagenomic sequence classification [17] highlight these limitations and underscore the need for more flexible and efficient solutions. Moreover, the need for uniform sequence lengths and expert-driven feature selection can delay critical responses during outbreaks.

Recent advancements in deep learning, particularly CNNs, offer promising solutions by transforming genetic sequences into image-based representations [18], [19]. These approaches enable handling complex and variable genetic data more effectively than traditional methods. However, previous studies have not focused on hydrophobicity signals, leaving a gap in leveraging this biologically relevant feature for genomic analysis. By integrating hydrophobicity profiles into the analysis, we can potentially improve the accuracy and efficiency of viral classification.

This study proposes a novel approach to viral classification by leveraging CNNs and hydrophobicity images derived from DNA sequences. By transforming genetic sequences into hydrophobicity-based visual representations, we enable CNNs to handle viral genetic data's inherent complexity and variability more effectively. This approach reduces the need for extensive preprocessing and fixed input lengths, providing a flexible and efficient solution for viral classification.

Our work presents several unique contributions that distinguish it from existing literature: “ i) innovative use of hydrophobicity signals: we introduce a novel method for representing genetic information using hydrophobicity signals derived from DNA sequences, a feature not explored by traditional nucleotide-based methods, ii) application of CNN: leveraging CNNs, we transform genetic sequences into hydrophobicity-based visual representations, enhancing the model's ability to process complex and variable viral genetic data, iii) improved flexibility and efficiency: our method addresses the limitations of conventional sequence-based models, such as the requirement for fixed sequence lengths and extensive feature extraction, offering a more adaptable and efficient approach to viral classification.

The remainder of this paper is structured as follows: section 2 details the methodology, including data collection, preparation, model implementation, and experimental setup. Section 3 presents the results and discussion, comparing the performance of our CNN-based approach with baseline machine learning models and previous studies. Finally, Section 4 concludes the paper and suggests future research directions.

## 2. METHOD

This section outlines the data collection, preprocessing, model implementation, and experimental setup used in this study. The focus is on classifying SARS-CoV-2 and non-SARS-CoV-2 viruses using convolutional neural networks (CNN) with hydrophobicity signals derived from DNA sequences. The approach addresses the limitations of conventional methods that require fixed-length sequences and extensive feature extraction preprocessing.

### 2.1. Data collection

We collected DNA sequences of seven coronaviruses, categorized into SARS-CoV-2 and non-SARS-CoV-2 classes. These sequences were obtained from the NCBI virus database by searching for the desired virus type at [www.ncbi.nlm.nih.gov/labs/virus](http://www.ncbi.nlm.nih.gov/labs/virus). The dataset comprises 8,143 samples, with 4,156 classified as non-SARS-CoV-2 and 3,987 as SARS-CoV-2. These sequence lengths ranged from 8 to 31,104

base pairs (bps). Detailed information, including the number of sequences and their length ranges for each virus type, is summarized in Table 1.

Table 1. Detailed description of the coronavirus dataset

| Names      | Numbers of sequences | Min. length | Max. length | Class |
|------------|----------------------|-------------|-------------|-------|
| HCoV-OC43  | 1,149                | 8           | 31,104      | 0     |
| HCoV-229E  | 619                  | 20          | 28,754      | 0     |
| HCoV-HKU1  | 412                  | 81          | 30,144      | 0     |
| HCoV-NL63  | 669                  | 81          | 27,833      | 0     |
| MERS-CoV   | 1,259                | 110         | 30,484      | 0     |
| SARS-CoV   | 18                   | 158         | 29,751      | 0     |
| SARS-CoV-2 | 3,987                | 278         | 29,909      | 1     |

## 2.2. Data preparation

The DNA sequences, consisting of the nucleotide bases adenine (A), cytosine (C), guanine (G), and thymine (T), were translated into amino acid sequences using the standard genetic code. This translation involves reading the DNA sequence in triplets of nucleotides, known as codons, each corresponding to a specific amino acid or a stop signal [20]. Given a DNA sequence  $d_i = \{b_1, b_2, \dots, b_{n_i}\}$  where  $b_j$  represents the  $j$ -th nucleotide, the translation function  $T$  converts every triplet  $(b_{3k-2}, b_{3k-1}, b_{3k})$  into an amino acid  $a_k$  until the end of the sequence is reached. If  $n_i$  is not a multiple of 3, the remaining nucleotides are ignored. Mathematically, the translation can be described as in (1):

$$a_i = T(d_i) = \left\{ T(b_{3k-2}, b_{3k-1}, b_{3k}) \mid k = 1, 2, \dots, \left\lfloor \frac{n_i}{3} \right\rfloor \right\} \quad (1)$$

where  $T$  is the mapping function from codons to amino acids based on the standard genetic code.

Hydrophobicity profiles were generated for each amino acid sequence to create image-based representations. We employed the Kyte-Doolittle scale [21], a widely used measure for hydrophobicity, to quantify the hydrophobicity of amino acid residues. Mathematically, the hydrophobicity value  $H$  for each amino acid  $i$  is represented as  $H_i$ , where  $i$  denotes the position in the amino acid sequence. Given an amino acid sequence  $\{A_1, A_2, \dots, A_n\}$ , where  $n$  is the length of the sequence, each amino acid  $A_i$  is transformed into its corresponding hydrophobicity value  $H_i$  using the Kyte-Doolittle scale. The hydrophobicity profile for the entire sequence can be represented in (2):

$$H = \{H_1, H_2, \dots, H_n\} \quad (2)$$

To convert the linear hydrophobicity profile into a 2D image, we first define a window size  $w$ . This window size determines the dimensions of the resulting image, where each pixel intensity corresponds to the average hydrophobicity value within the window. For an image of dimension  $m \times m$ , we reshaped the sequence into a matrix format. The value of  $m$  is chosen based on the sequence length  $n$  and the desired resolution. The transformation process involves the following steps:

- Windowing: divide the hydrophobicity sequence  $H$  into overlapping or non-overlapping windows of size  $w$ . For  $j = 1, 2, \dots, \left\lfloor \frac{n}{w} \right\rfloor$ , each window  $W_j$  is defined as in (3):

$$W_j = \{H_{(j-1)w+1}, H_{(j-1)w+2}, \dots, H_{jw}\} \quad (3)$$

- Averaging: calculate the average hydrophobicity value for each window  $W_j$ . The average hydrophobicity value  $\bar{H}_j$  for the window  $W_j$  is given by (4):

$$\bar{H}_j = \frac{1}{w} \sum_{k=(j-1)w+1}^{jw} H_k \quad (4)$$

- Matrix formation: arrange the average hydrophobicity values  $\bar{H}_j$  into a matrix  $M$  of dimension  $m \times m$ . The matrix  $M$  is formed in (5) as follows:

$$M = \begin{pmatrix} \bar{H}_1 & \bar{H}_2 & \dots & \bar{H}_m \\ \bar{H}_{m+1} & \bar{H}_{m+2} & \dots & \bar{H}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{H}_{m(m-1)+1} & \bar{H}_{m(m-1)+2} & \dots & \bar{H}_{m^2} \end{pmatrix} \tag{5}$$

- Normalization: normalize the matrix M to ensure the pixel values are within a specific range (e.g., 0 to 255 for grayscale images). The normalized matrix M is given by (6):

$$M' = 255 \times \frac{M - \min(M)}{\max(M) - \min(M)} \tag{6}$$

- Image representation: the normalized matrix M' is then converted into a grayscale image, where each pixel intensity represents the hydrophobicity value of the corresponding window in the sequence.
- Remove non-essential graphical elements: to prepare clean images, labels, titles, legends, and axes were removed.

By transforming the amino acid sequences into hydrophobicity-based images following (2)-(6), we enable the CNN to effectively capture the spatial patterns and characteristics inherent in the hydrophobicity profiles. This transformation enhanced the model's ability to classify SARS-CoV-2 and non-SARS-CoV-2 viruses. Figure 1 illustrates the process of generating hydrophobicity signals from DNA sequences.

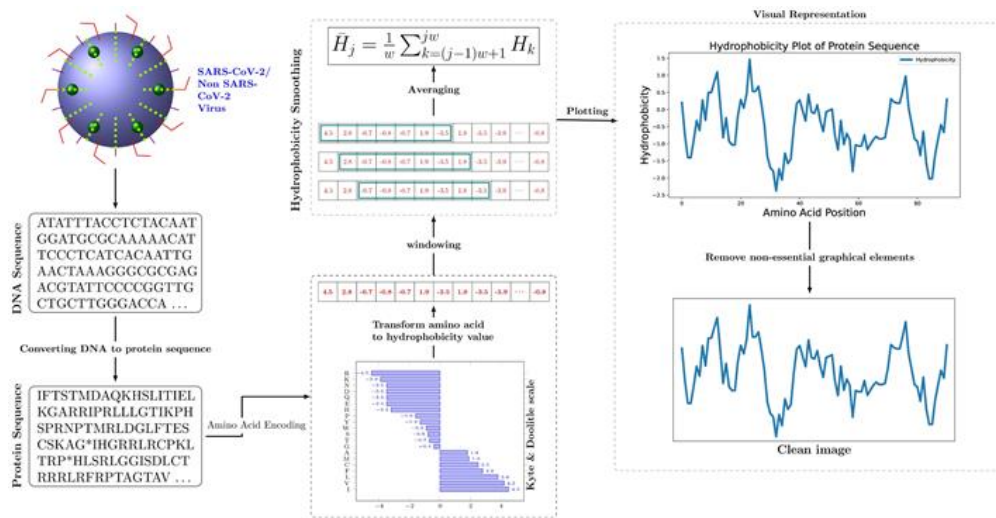


Figure 1. Preprocessing workflow: DNA sequence to hydrophobicity signal conversion

### 2.3. Model implementation

We implemented a convolutional neural network (CNN) to process image-based hydrophobicity profiles. The CNN architecture includes multiple convolutional, max-pooling, and fully connected layers. This structure allows the model to capture complex patterns within the hydrophobicity images, improving its performance in the binary classification. The network begins with input images of hydrophobicity profiles, which are processed through convolutional and pooling layers to extract relevant features. These extracted features are passed through fully connected layers to generate the final binary classification output. The architecture of the CNN is depicted in Figure 2.

Table 2 provides the details of CNN architecture. Each layer's type, the number of filters or units, kernel size, activation function, and output shape are specified to give a comprehensive overview of the network structure. This detailed breakdown helps us understand the flow of data through the network and the transformations applied at each stage.

In the first layer, a convolutional operation is applied with 32 filters of size 3×3, producing an output shape of (128, 128, 32). This convolutional layer is followed by a max-pooling layer with a 2×2 filter size, which reduces the spatial dimensions to (64, 64, 32). The second convolutional layer uses 64 filters of size 3×3, resulting in an output shape of (64, 64, 64). This layer is then followed by another max-pooling layer, reducing the dimensions to (32, 32, 64). The third convolutional layer applies 128 filters of size 3×3, producing an output shape of (32, 32, 128). It is followed by a max-pooling layer, which further reduces the

dimensions to (16, 16, 128). These convolutional and pooling layers are designed to extract and condense features progressively, capturing the essential patterns in the hydrophobicity profiles. The output of the final pooling layer is then flattened and fed into a fully connected dense layer with 512 units, which provides a rich representation of the extracted features. This dense layer is crucial for integrating the features obtained by the previous convolutional layers and preparing them for classification. The final layer is another dense layer equipped with a sigmoid activation function, which produces a probability for the binary classification (SARS-CoV-2 vs. non-SARS-CoV-2). The output shape of this final layer is (1), corresponding to the binary classification task. This setup allows the model to output a probability for the positive class, facilitating accurate classification. The CNN was implemented using the TensorFlow and Keras libraries, allowing efficient model training and evaluation.

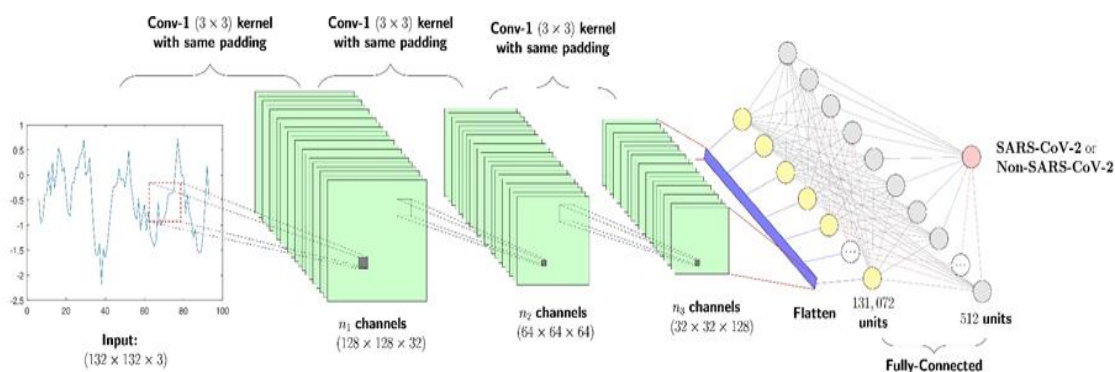


Figure 2. The architecture of the CNN model: layer configuration and sequence

Table 2. CNN model specification

| Layer | Type                    | Number of filters | Kernel size | Output shape   |
|-------|-------------------------|-------------------|-------------|----------------|
| 1     | Convolution             | 32                | 3×3         | (128, 128, 32) |
| 2     | Max pooling             | -                 | 2×2         | (64, 64, 32)   |
| 3     | Convolution             | 64                | 3×3         | (64, 64, 64)   |
| 4     | Max pooling             | -                 | 2×2         | (32, 32, 64)   |
| 5     | Convolution             | 128               | 3×3         | (32, 32, 128)  |
| 6     | Max pooling             | -                 | 2×2         | (16, 16, 128)  |
| 7     | Fully connected (dense) | 512               | -           | (512)          |
| 8     | Output (dense, sigmoid) | 1                 | -           | (1)            |

We include a variety of baseline machine learning models implemented using the sci-kit-learn library, which is compared against the CNN model. These models are explicitly configured for handling DNA sequence data as follows:

- Support vector machine (SVM): utilizes a linear kernel to classify linearly separable data. Non-linear classification is enabled using kernel tricks, with the regularization parameter set to 1.0, balancing the classification margin with error minimization [22].
- K-nearest neighbors (K-NN): employ a distance metric (Euclidean metric) to identify the closest training examples and determine the classification [23]. Set the number of neighbors as 3 to ensure proximity affects the classification decision.
- Logistic regression (LR): predicts categorical outcomes using a logistic function, modeling probability distributions [24]. It was configured with a 'linear' solver, a regularization strength of 1, and an L2 penalty to mitigate overfitting.
- Decision tree (DT): this model classifies data by creating a tree that models decisions based on feature values. It is configured without a maximum depth to allow detailed tree growth and uses a minimum sample split of 2 [25].
- Random forest (RF): this ensemble model builds multiple decision trees and merges their output to improve accuracy and control overfitting. It used 100 estimators and the 'sqrt' method for feature selection, optimizing variance reduction and predictive accuracy [26].
- XGBoost: an implementation of gradient-boosted decision trees designed for speed and performance, configured with 100 estimators, a maximum depth of 3, and a learning rate 0.1. Grid search is used to fine-tune parameters based on performance [27].

- Multilayer perceptron (MLP): a neural network model consisting of multiple layers of nodes, including an input layer, one or more hidden layers, and an output layer [28]. It was tuned with a maximum iteration of 100 to ensure convergence and optimized for binary classification tasks.

#### 2.4. Experimental setup

This phase outlines the experimental setup used to evaluate the performance of the proposed CNN model against various baseline models. We assessed model performance using key metrics: accuracy, precision, recall, F1-score, and AUC-ROC. Additionally, we measured the computation time required for training and prediction to evaluate the practicality and efficiency of the method. The experiment was structured as follows:

- Comparison with baseline methods using matching sequence lengths: each model was trained on the same dataset and divided into training, validation, and testing sets in a 70-15-15 ratio. Uniform data preprocessing steps, such as normalization and encoding, were applied before training.
- Comparison with baseline methods using extended sequence lengths: test data included sequence lengths exceeding the maximum sequence length of the training data to test the models' stability and robustness with longer sequences.
- Assessment of computation time: to evaluate the method's practicality and efficiency, we measured the time required to train the model, process raw sequences into images, and predict new data.
- Confusion matrix and AUC-ROC analysis: we analyzed the confusion matrix and ROC curve with the AUC score to visualize true positives, false positives, and false negatives, providing insight into classification accuracy and the model's discriminative ability.

We employed a validation strategy to enhance model robustness and reliability, including using a validation set to fine-tune the models and prevent overfitting. Additionally, we utilized 10-fold cross-validation, dividing the training dataset into ten subsets. The model was trained ten times, each iteration using a different subset for validation, ensuring enhanced generalizability.

##### 2.4.1. Computational resources

In the experiments, we used an Intel Xeon CPU, an NVIDIA Tesla P100 GPU with 16 GB of memory, and 29 GB of RAM. We conducted all experiments using Kaggle Notebooks with Python 3.7. We used TensorFlow 2.1, scikit-learn, NumPy, and Pandas for machine learning. This setup handled the computational demands of processing large datasets and training complex models such as CNN. Kaggle provided a stable platform for reliable execution, and the high-performance GPU efficiently processed computationally heavy tasks, such as analyzing DNA sequences.

##### 2.4.2. Data availability

We have made the dataset used in this study available to the public on Kaggle to ensure accessibility for further research. To access the raw data, please visit raw-dataset. For data transformed into a hydrophobicity signal, please refer to the Image-dataset. These datasets are provided under an open-access license, encouraging further research and validation of the findings.

### 3. RESULTS AND DISCUSSION

In this section, we presented and organized the experimental results into five parts: i) a comparison of the proposed method with baseline methods using test data with sequence lengths matching those in the training data, ii) an evaluation of the method against baseline methods using test data with sequence lengths exceeding the maximum length of the training data, iii) an assessment of the computation time required to train the model, process raw sequences into images, and make predictions on new data, iv) an analysis of the prediction results using a confusion matrix and AUC-ROC score, and v) a comparison with previous studies. The objective is to evaluate the performance of the proposed model using various metrics under different conditions and to assess the practicality and efficiency of the method in real-world applications.

#### 3.1. Comparison with baseline models using matching sequence lengths

We compared the proposed model's performance with various baseline methods using test data where the sequence lengths matched those in the training data. The proposed method employs a CNN with hydrophobicity signals as features, while the baseline methods use raw sequences. To evaluate the effectiveness of each model, we focused on key metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. Table 3 provides a comparative performance summary of classification models using these two types of features.

The results indicate that the baseline methods performed exceptionally well, achieving average accuracy, precision, recall, F1-score, and AUC-ROC of 98.91%, 98.28%, 99.52%, 98.89%, and 99.71%, respectively. The proposed model significantly improved over the baseline methods, with an increase of

approximately 0.8% across all metrics. In this scenario, all methods effectively classified the data, maintaining an average error rate of less than 0.1%.

Table 3. Performance summary of classifiers with test sequence lengths within the training range

| Classifier | Feature Type          | Acc.   | Prec.  | Rec.   | F1     | AUC-ROC |
|------------|-----------------------|--------|--------|--------|--------|---------|
| SVM        | DNA Sequence          | 0.9910 | 0.9819 | 1.0000 | 0.9909 | 0.9998  |
| KNN        | DNA Sequence          | 0.9836 | 0.9817 | 0.9849 | 0.9833 | 0.9954  |
| LR         | DNA Sequence          | 0.9902 | 0.9835 | 0.9967 | 0.9900 | 0.9982  |
| DT         | DNA Sequence          | 0.9869 | 0.9850 | 0.9883 | 0.9866 | 0.9869  |
| RF         | DNA Sequence          | 0.9894 | 0.9787 | 1.0000 | 0.9892 | 0.9997  |
| XGBoost    | DNA Sequence          | 0.9910 | 0.9819 | 1.0000 | 0.9909 | 0.9997  |
| MLP        | DNA Sequence          | 0.9918 | 0.9868 | 0.9967 | 0.9917 | 0.9997  |
| Ours       | hydrophobicity signal | 0.9984 | 0.9967 | 1.0000 | 0.9983 | 0.9999  |

The CNN model achieved the highest accuracy at 99.84%, indicating its superior ability to correctly classify positive and negative cases compared to the baseline models. With a high precision of 99.67%, the model makes very few false positive errors, which is crucial for avoiding the misclassification of non-SARS-CoV-2 cases as SARS-CoV-2. The model also achieved perfect recall at 100%, meaning it correctly identifies all actual positive cases, ensuring no SARS-CoV-2 cases are missed. The F1 Score of 99.83% demonstrates the model's overall effectiveness by balancing precision and recall. Furthermore, the AUC-ROC score of 0.9999 illustrates the model's excellent performance in distinguishing between classes across different threshold values, highlighting its discriminative power.

The significant improvement across all metrics highlights the CNN model's robustness in handling the variability and complexity of viral genetic material. The use of a hydrophobicity signal allows the CNN model to capture intricate patterns in the data that traditional sequence-based methods do not utilize as effectively. The high-performance metrics suggest that the CNN model is reliable and could be used in real-world applications for rapid and accurate viral classification, potentially leading to better outbreak management and public health responses.

### 3.2. Performance evaluation with extended sequence lengths

We evaluated the robustness and adaptability of the proposed method by comparing it with baseline methods using test data with sequence lengths exceeding the maximum length of the training data. The analysis focused on how well the models handled longer sequences and the impact on performance metrics. A two-phase study was conducted to assess the robustness of the CNN classifier using hydrophobicity signals. In the first phase, we used raw DNA sequences as inputs for baseline classifiers and measured their performance. In the second phase, we used hydrophobicity signals as inputs for all classifiers to compare their performance directly against the baseline methods.

#### 3.2.1. Performance comparison: baseline classifier with DNA sequence vs. CNN with hydrophobicity signal

In this phase, the test set sequences were longer than those in the training set. This scenario aimed to determine whether the classification models could predict the test set as effectively as in the first scenario. Table 4 presents a performance comparison of various classification models on an extended test set where the DNA sequence length is longer than that of the training set.

The experimental results show that the baseline methods achieved much lower average accuracy, precision, recall, F1-score, and AUC-ROC compared to the first experiment, specifically 44.36%, 91.23%, 44.36%, 49.40%, and 96.29%, respectively. Meanwhile, the proposed method maintained high performance, achieving more than 99.73% for all metrics. This result demonstrates the robustness of our proposed method in handling differences in data sample size during the prediction phase, even when the data size is outside the range of the training set. The CNN model's high performance can be attributed to its ability to capture intricate patterns in the hydrophobicity signal derived from DNA sequences. Unlike traditional methods that rely on direct sequence analysis, the image-based approach leverages the spatial information in the hydrophobicity profiles, allowing the model to learn complex features that enhance classification accuracy.

Furthermore, while precision and AUC-ROC scores were high for the baseline methods, exceeding 90%, the other metrics did not exceed 50%. These experimental results indicate that the baseline models are very conservative in predicting the positive class, resulting in high precision. They only predict positives when they are very confident, leading to few false positives. However, this conservatism results in many actual positives being missed (high false negatives), which lowers recall. The high AUC-ROC value indicates that the models are good at distinguishing between positive and negative classes. Still, they struggle with selecting a threshold that balances precision and recall. The low accuracy suggests the prediction imbalance leads to poor performance regarding correct classifications. In summary, these metrics indicate that the

baseline models are cautious and only predict positive when very confident, leading to high precision and AUC-ROC but at the cost of recall and overall accuracy. Our proposed method, by contrast, shows significant robustness and adaptability, making it well-suited for real-world applications where sequence lengths can vary significantly.

Table 4. Comparative performance metrics for extended test sets with longer sequence lengths: baseline

| DNA sequence vs. CNN hydrophobicity signal |                       |        |        |        |        |         |
|--|-----------------------|--------|--------|--------|--------|---------|
| Classifier                                 | Feature Type          | Acc.   | Prec.  | Rec.   | F1     | AUC-ROC |
| SVM  | DNA Sequence          | 0.2550 | 0.9072 | 0.2550 | 0.2789 | 0.9988  |
| KNN  | DNA Sequence          | 0.5048 | 0.9127 | 0.5048 | 0.5833 | 0.9745  |
| LR   | DNA Sequence          | 0.5723 | 0.9150 | 0.5723 | 0.6480 | 0.9461  |
| DT   | DNA Sequence          | 0.6976 | 0.9215 | 0.6976 | 0.7557 | 0.8309  |
| RF   | DNA Sequence          | 0.2305 | 0.9068 | 0.2305 | 0.2414 | 0.9924  |
| XGBoost                                    | DNA Sequence          | 0.2691 | 0.9074 | 0.2691 | 0.2996 | 0.9980  |
| MLP  | DNA Sequence          | 0.5758 | 0.9152 | 0.5758 | 0.6512 | 0.9998  |
| Ours                                       | hydrophobicity signal | 0.9991 | 1.0000 | 0.9917 | 0.9959 | 1.0000  |

### 3.2.2. Performance comparison for all classifiers using hydrophobicity signal as input features

In this phase, we compare the performance of various classifiers using hydrophobicity signals as input features. Hydrophobicity signals are input for the convolutional neural network and baseline machine learning models. The comparison focuses on key metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to evaluate the effectiveness of each classifier. The results of our experiments are summarized in Table 5, providing a comprehensive evaluation of each model's performance.

Table 5. Comparative performance of classifiers using hydrophobicity signal on extended test sets

| Classifier | Feature Type          | Acc.   | Prec.  | Rec.   | F1     | AUC-ROC |
|------------|-----------------------|--------|--------|--------|--------|---------|
| SVM        | hydrophobicity signal | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000  |
| KNN        | hydrophobicity signal | 0.9667 | 0.7610 | 1.0000 | 0.8643 | 0.9980  |
| LR         | hydrophobicity signal | 0.9930 | 0.9449 | 0.9917 | 0.9677 | 0.9970  |
| DT         | hydrophobicity signal | 0.8510 | 0.4158 | 1.0000 | 0.5874 | 0.9167  |
| RF         | hydrophobicity signal | 0.9974 | 0.9836 | 0.9917 | 0.9877 | 0.9996  |
| XGBoost    | hydrophobicity signal | 0.9912 | 0.9302 | 0.9917 | 0.9600 | 0.9970  |
| MLP        | hydrophobicity signal | 0.9974 | 0.9836 | 0.9917 | 0.9877 | 0.9975  |
| Ours       | hydrophobicity signal | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000  |

The baseline methods exhibited a range of performance outcomes when using hydrophobicity signals as input features. The support vector machine (SVM) achieved perfect scores across all metrics, demonstrating its exceptional classification capability with hydrophobicity signals. Our proposed method matched this performance, also achieving perfect scores, underscoring its robustness and effectiveness. The K-nearest neighbors (KNN) model showed an impressive recall of 100.00%, highlighting its ability to identify all true positive cases. However, its lower precision (76.10%) and F1-score (86.43%) indicate a higher incidence of false positives, which impacts its overall reliability. Logistic regression (LR) displayed robust performance with a 99.30% accuracy and a 96.77% F1-score, indicating a well-balanced model that effectively manages precision and recall. This result makes LR a reliable choice for classification tasks. Random forest (RF) and XGBoost also performed admirably, with accuracies of 99.74% and 99.12%, respectively, and high F1 scores. While slightly trailing our proposed method, their strong performance indicates their suitability for handling hydrophobicity signals. Decision tree (DT) faced significant challenges, evidenced by its lower accuracy of 85.10% and substantial drops in precision (41.58%) and F1-score (58.74%). These results highlight the difficulty DT models face in managing the complexity of hydrophobicity signal data. The multilayer perceptron (MLP) performed significantly better than previously noted, with an accuracy of 99.74%, precision of 98.36%, recall of 99.17%, and an F1-score of 98.77%. This substantial improvement positions MLP as a strong contender among the classifiers, though still slightly behind the perfect scores achieved by SVM and our proposed method.

Overall, the variability in the performance of the baseline methods highlights each model's unique strengths and weaknesses. The perfect scores achieved by our proposed method and SVM emphasize the robustness and reliability of these approaches in using hydrophobicity signals for viral classification. The findings underscore the importance of selecting the appropriate classifier for specific tasks, with our proposed method offering a well-balanced and efficient solution for handling extended test sets.



### 3.3. Computation time assessment

This subsection evaluates the computation time required to train the models, process raw sequences into images, and make predictions on new data. The analysis aims to assess the practicality and efficiency of our proposed method in real-world applications. To gauge computational efficiency, we measured the time needed for various process stages, including model training, data preprocessing, and prediction.

We recorded the time required to train the CNN model using a hydrophobicity signal and compared it to the time needed for training baseline models using raw DNA sequences. Our findings indicated that the CNN model demanded more computational resources and longer training times due to the complexity of processing image data. However, the increased training time is justified by the significantly higher performance metrics the CNN model achieves. We also measured the preprocessing time necessary to convert raw DNA sequences into hydrophobicity signals. This process involves translating DNA into amino acid sequences, calculating hydrophobicity profiles, and generating the corresponding images. Although computationally intensive, this preprocessing step provides a robust feature representation that significantly enhances the model's overall performance. Finally, we assessed the time required to make predictions on new data. The CNN model demonstrated efficient prediction times comparable to the baseline models. Despite the initial preprocessing overhead, the CNN model's prediction phase was optimized to handle image data swiftly, ensuring practical applicability in real-time scenarios. Table 6 summarizes the computational efficiency metrics for both the CNN and baseline models:

**Table 6. Computational efficiency comparison: training and prediction times**

| Classifier | Feature type          | Training time (sec) | Prediction time (sec/sample) |
|------------|-----------------------|---------------------|------------------------------|
| SVM        | hydrophobicity signal | 718.0               | 0.30                         |
| KNN        | hydrophobicity signal | 0.3                 | 0.10                         |
| LR         | hydrophobicity signal | 33.5                | 0.20                         |
| DT         | hydrophobicity signal | 74.0                | 0.50                         |
| RF         | hydrophobicity signal | 23.2                | 0.10                         |
| XGBoost    | hydrophobicity signal | 674.0               | 0.15                         |
| MLP        | hydrophobicity signal | 39.1                | 0.12                         |
| Ours       | hydrophobicity signal | 58.6                | 0.30                         |

The results in Table 6 indicate that while our proposed CNN model required a moderate training time (58.6 seconds) compared to some baseline models, it offered efficient prediction times (0.30 seconds per sample) comparable to those of the other methods. Specifically, SVM and XGBoost required significantly longer training times (718.0 seconds and 674.0 seconds, respectively) but demonstrated comparable prediction times (0.30 and 0.15 seconds per sample). KNN, RF, and MLP had notably shorter training times (0.3, 23.2, and 39.1 seconds, respectively) and faster prediction times (0.10, 0.10, and 0.12 seconds per sample), highlighting their computational efficiency. Logistic regression (LR) and decision tree (DT) showed intermediate training times (33.5 and 74.0 seconds, respectively) and varied prediction times (0.20 and 0.50 seconds per sample), reflecting their balanced computational demands. Despite the initial preprocessing overhead, the CNN model's optimized prediction phase ensures practical applicability in real-time scenarios. The results suggest that the initial investment in training our proposed CNN model is justified by its robust performance and efficient prediction capabilities in real-world applications.

### 3.4. Confusion matrix and AUC-ROC analysis of prediction results

To provide a detailed view of our proposed method's performance, we present the confusion matrix for the prediction results from the first experiment in Table 3. The confusion matrix helps visualize the true positives, false positives, and false negatives, providing insight into the classification accuracy. The confusion matrix and ROC curve for our proposed method are shown in Figure 3.

From the confusion matrix in Figure 3(a), we observe the following: true positives (TP): 622, true negatives (TN): 598, false positives (FP): 2, and false negatives (FN): 0. The high number of true positives and true negatives, along with the low number of false positives and false negatives, indicates that our proposed method achieves high accuracy in classifying the test data. These results further reinforce the effectiveness and reliability of our CNN-based approach using hydrophobicity signal.

In addition to the confusion matrix, we present the receiver operating characteristic (ROC) curve and the area under the curve (AUC) score to illustrate the model's performance. The ROC curve in Figure 3(b) demonstrates our model's excellent performance, with an AUC-ROC score of 99.99%. This near-perfect score indicates that the proposed method is highly discriminative, distinguishing between positive and negative classes.

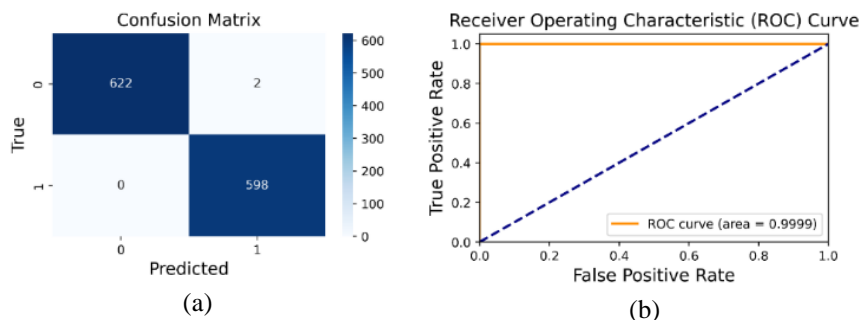


Figure 3. Proposed method (a) confusion matrix and (b) ROC curve analysis for the CNN

### 3.5. Comparison with previous studies

We compared this study's results with three prior studies that employed machine learning and deep learning techniques to classify viral genomes, specifically on SARS-CoV-2. Randhawa *et al.* [29] utilized k-mers and machine learning algorithms, achieving an accuracy of 100%. However, the lack of reported hyperparameters and the small sample size in their work may have led to overfitting, thus compromising the reproducibility and generalizability of their results. Additionally, they reported a computation time of 2.14 seconds per classification, significantly higher than our proposed method. Lopez-Rincon *et al.* [18] employed deep neural networks (DNNs), achieving an accuracy of 98.73%. Despite the high accuracy, their study was limited by an imbalanced dataset and the exclusion of a control group, which may affect the model's generalizability.

Furthermore, the DNN approach is computationally intensive, requiring substantial data and computational resources. Singh *et al.* [22] Adopting a digital signal processing (DSP) and machine learning (ML) approach, they achieved a sensitivity of 96.29%, specificity of 98.25%, and accuracy of 97.47%. Their method was notably efficient, with a computation time of 0.31 seconds per classification. However, their study's reliance on a small number of samples and partial coding sequences (CDS) may affect the robustness of their results.

Our proposed CNN-based method using hydrophobicity signal demonstrated superior performance, achieving over 99.91% accuracy, precision, recall, F1-score, and AUC-ROC. The method showed robustness in handling longer test sequences, maintaining high performance across all metrics. Additionally, the computation time for our method was only 0.3 seconds per classification, making it suitable for real-time applications. Our approach also ensured balanced performance across all metrics, indicating a well-generalized model capable of flexible input handling due to the use of a hydrophobicity signal. A comparative summary is provided in Table 7.

Despite the promising results, our study has some limitations. The computational cost of generating a hydrophobicity signal and training the CNN model is high, which may limit its scalability. Additionally, the model's performance was not validated with biological insights or experimental data, which would strengthen its applicability in real-world virus classification scenarios.

Table 7. Comparative analysis of SARS-CoV-2 classification methods

| Study                      | Method                   | Acc.    | Prec.   | Rec.   | F1     | CT (sec) |
|----------------------------|--------------------------|---------|---------|--------|--------|----------|
| Randhawa <i>et al.</i>     | k-mers + ML              | 100.00% | -       | -      | -      | 2.14     |
| Lopez-Rincon <i>et al.</i> | DNN                      | 98.73%  | -       | -      | -      | -        |
| Singh <i>et al.</i>        | DSP + ML (RF)            | 97.47%  | 96.29%  | 98.25% | -      | 0.31     |
| Our Method                 | CNN + Hydrophobicity Img | 99.91%  | 100.00% | 99.17% | 99.59% | 0.3      |

## 4. CONCLUSION

In this work, we have demonstrated the robustness and effectiveness of our proposed classification model, which utilizes a hydrophobicity signal derived from DNA sequences to distinguish between SARS-CoV-2 and non-SARS-CoV-2 viruses. This classification is crucial for understanding viral genomes and has significant implications for drug discovery and public health. Our experimental results show that our model consistently outperforms baseline methods that use raw sequence data, particularly when handling test sequences that differ in length from those in the training set. Our proposed method achieved over 99% accuracy, precision, recall, F1-score, and AUC-ROC. This performance is significantly higher than the baseline methods, which struggled with longer sequences and attained much lower average metrics. These results highlight the potential of hydrophobicity signals as a superior feature representation, providing a more

consistent and informative input for classification tasks. Despite the baseline methods' low recall and accuracy, their high precision and AUC-ROC values indicate a conservative approach to positive class prediction, resulting in many false negatives. In contrast, our model's balanced performance across all metrics suggests optimal threshold selection and comprehensive generalization ability across different data samples. This balanced approach addresses concerns about overfitting and enhances reliability in diverse scenarios. While our method shows great promise, future work should focus on optimizing computational efficiency and exploring additional feature extraction techniques to improve performance further. Additionally, validating our model's predictions with biological insights and experimental data will strengthen its applicability in real-world virus classification scenarios. This continuous improvement will ensure the model remains robust and reliable for rapid virus detection, ultimately contributing to more effective public health interventions and bioinformatics applications.

## ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support provided by the Ministry of Research, Technology, and Higher Education, Republic of Indonesia, under contract number 1928/PKS/ITS/2023, which has made this research possible.




## REFERENCES

- [1] B. B. Oude Munnink *et al.*, "Rapid sars-cov-2 whole-genome sequencing and analysis for informed public health decision-making in the netherlands," *Nature Medicine*, vol. 26, no. 9, pp. 1405–1410, 2020, doi: 10.1038/s41591-020-0997-y.
- [2] N. C. Stenseth, G. Dharmarajan, R. Li, Z. L. Shi, R. Yang, and G. F. Gao, "Lessons learnt from the covid-19 pandemic," *Frontiers in Public Health*, vol. 9, 2021, doi: 10.3389/fpubh.2021.694705.
- [3] C. N. Pace *et al.*, "Contribution of hydrophobic interactions to protein stability," *Journal of Molecular Biology*, vol. 408, no. 3, pp. 514–528, May 2011, doi: 10.1016/j.jmb.2011.02.053.
- [4] T. Ohmura, T. Ueda, K. Ootsuka, M. Saito, and T. Imoto, "Stabilization of hen egg white lysozyme by a cavity-filling mutation," *Protein Science*, vol. 10, no. 2, pp. 313–320, 2001, doi: 10.1110/ps.37401.
- [5] U. Shekhawat and A. Roy Chowdhury (Chakravarty), "Computational and comparative investigation of hydrophobic profile of spike protein of sars-cov-2 and sars-cov," *Journal of Biological Physics*, vol. 48, no. 4, 2022, doi: 10.1007/s10867-022-09615-x.
- [6] H. X. Zhou and X. Pang, "Electrostatic interactions in protein structure, folding, binding, and condensation," *Chemical Reviews*, vol. 118, no. 4, pp. 1691–1741, 2018, doi: 10.1021/acs.chemrev.7b00305.
- [7] F. C. L. Almeida, K. Sanches, R. Pinheiro-Aguiar, V. S. Almeida, and I. P. Caruso, "Protein surface interactions—theoretical and experimental studies," *Frontiers in Molecular Biosciences*, vol. 8, 2021, doi: 10.3389/fmolb.2021.706002.
- [8] T. Dannenhoffer-Lafage and R. B. Best, "A data-driven hydrophobicity scale for predicting liquid-liquid phase separation of proteins," *Journal of Physical Chemistry B*, vol. 125, no. 16, pp. 4046–4056, 2021, doi: 10.1021/acs.jpcc.0c11479.
- [9] A. Telenti, E. B. Hodcroft, and D. L. Robertson, "The evolution and biology of sars-cov-2 variants," *Cold Spring Harbor Perspectives in Medicine*, vol. 12, no. 5, 2022, doi: 10.1101/cshperspect.a041390.
- [10] R. Rangan *et al.*, "RNA genome conservation and secondary structure in sars-cov-2 and sars-related viruses: a first look," *Rna*, vol. 26, no. 8, pp. 937–959, 2020, doi: 10.1261/RNA.076141.120.
- [11] H. Jiang *et al.*, "Evaluation of the inhibition potency of nirmatrelvir against main protease mutants of sars-cov-2 variants," *Biochemistry*, vol. 62, no. 13, pp. 2055–2064, 2023, doi: 10.1021/acs.biochem.3c00075.
- [12] M. Hussein *et al.*, "Efficient crispr-cas13d-based antiviral strategy to combat sars-cov-2," *Viruses*, vol. 15, no. 3, 2023, doi: 10.3390/v15030686.
- [13] T. I. Ng *et al.*, "Antiviral drug discovery for the treatment of covid-19 infections," *Viruses*, vol. 14, no. 5, 2022, doi: 10.3390/v14050961.
- [14] M. S. Mottaqi, F. Mohammadipناه, and H. Sajedi, "Contribution of machine learning approaches in response to sars-cov-2 infection," *Informatics in Medicine Unlocked*, vol. 23, 2021, doi: 10.1016/j.imu.2021.100526.
- [15] A. S. Albahri *et al.*, "Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (covid-19): a systematic review," *Journal of Medical Systems*, vol. 44, no. 7, 2020, doi: 10.1007/s10916-020-01582-x.
- [16] S. Tiwari, P. Chanak, and S. K. Singh, "A review of the machine learning algorithms for covid-19 case analysis," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 1, pp. 44–59, 2023, doi: 10.1109/TAI.2022.3142241.
- [17] Z. Bzhalava, A. Tampuu, P. Bala, R. Vicente, and J. Dillner, "Machine learning for detection of viral sequences in human metagenomic datasets," *BMC Bioinformatics*, vol. 19, no. 1, 2018, doi: 10.1186/s12859-018-2340-x.
- [18] A. Lopez-Rincon *et al.*, "Classification and specific primer design for accurate detection of sars-cov-2 using deep learning," *Scientific Reports*, vol. 11, no. 1, 2021, doi: 10.1038/s41598-020-80363-5.
- [19] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary Intelligence*, vol. 15, no. 1, 2022, doi: 10.1007/s12065-020-00540-3.
- [20] M. H. Saier, "Understanding the genetic code," *Journal of Bacteriology*, vol. 201, no. 15, 2021, doi: 10.1128/JB.00091-19.
- [21] Jack Kyte and Russell F. Doolittle, "A simple method for displaying the hydropathic character of a protein," *Journal of Molecular Biology*, vol. 157, no. 1, pp. 105–132, 1982.
- [22] O. P. Singh, M. Vallejo, I. M. El-Badawy, A. Aysha, J. Madhanagopal, and A. A. Mohd Faudzi, "Classification of sars-cov-2 and non-sars-cov-2 using machine learning algorithms," *Computers in Biology and Medicine*, vol. 136, p. 104650, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104650.
- [23] H. Arslan and H. Arslan, "A new covid-19 detection method from human genome sequences using cpg island features and knn classifier," *Engineering Science and Technology, an International Journal*, vol. 24, no. 4, pp. 839–847, Aug. 2021, doi: 10.1016/j.jestech.2020.12.026.
- [24] M. Jamhuri, I. Mukhlash, and M. I. Irawan, "Performance improvement of logistic regression for binary classification by gaussian-newton method," in *2022 5th International Conference on Mathematics and Statistics*, 2022, doi: 10.1145/3545839.3545842.




- [25] S. H. Yoo *et al.*, “Deep learning-based decision-tree classifier for covid-19 diagnosis from chest x-ray imaging,” *Frontiers in Medicine*, vol. 7, Jul. 2020, doi: 10.3389/fmed.2020.00427.
- [26] 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, 2021, doi: 10.26599/BDMA.2020.9020016.
- [27] I. Muhammad, I. Mukhlash, M. Jamhuri, M. Iqbal, and M. I. Irawan, “Classification of covid-19 variants using boosting algorithm,” in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2022, pp. 29–34. doi: 10.23919/EECSI56542.2022.9946452.
- [28] A. Sarkar, J. Dey, and A. Bhowmik, “Multilayer neural network synchronized secured session key based encryption in wireless communication,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, pp. 169–177, 2019, doi: 10.11591/ijeecs.v14.i1.pp169-177.
- [29] G. S. Randhawa, M. P. M. Soltysiak, H. El Roz, C. P. E. de Souza, K. A. Hill, and L. Kari, “Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: covid-19 case study,” *PLoS ONE*, vol. 15, no. 4, 2020, doi: 10.1371/journal.pone.0232391.

## BIOGRAPHIES OF AUTHORS






**Mohammad Jamhuri**    is a lecturer at UIN Maulana Malik Ibrahim, Malang. He completed his undergraduate degree in Mathematics at the same university. He is currently pursuing his Doctorate in Mathematics at the Institut Teknologi Sepuluh Nopember (ITS), Surabaya, after completing his Master's degree there. His field of expertise is applied mathematics, machine learning, and numerical analysis. You can reach out to him via email: m.jamhuri@mat.uin-malang.ac.id.






**Mohammad Isa Irawan**    is a professor at the Institut Teknologi Sepuluh Nopember (ITS). He completed his Bachelor's degree in Mathematics at Universitas Airlangga. He then pursued his Master's in Engineering, specifically in the Control System group at the Department of Electrical Engineering at the Institut Teknologi Bandung. He earned his Doctorate in Philosophy, Software Engineering & Interactive System Group, Department of Informatics, from the Vienna University of Technology in Austria. His competencies include data science, machine learning, bioinformatics, decision support systems, and data mining. You can reach Mohammad Isa Irawan via email: mii@its.ac.id.



**Imam Mukhlash**    is a lecturer at the Institut Teknologi Sepuluh Nopember (ITS). He completed his Bachelor's degree in Mathematics at ITS. He pursued his Master's Degree and Doctorate in Data Mining at the Institut Teknologi Bandung (ITB), Bandung, Indonesia. He is associated with the Laboratory of Machine Learning and Big Data at ITS. His research interests include computational mathematics, data mining, artificial intelligence, and software engineering. He has made significant contributions to the field of machine learning and technology, with a focus on Machine Learning and Big Data. You can reach Dr. Imam Mukhlash via email: imamm@matematika.its.ac.id.



**Ni Nyoman Tri Puspaningsih**    is a professor at Universitas Airlangga. She completed her Bachelor's degree at Universitas Airlangga, her Master's degree at the Institut Teknologi Bandung, and her Doctorate at the Institut Pertanian Bogor. She also completed a Postdoctoral Program at the University of Groningen, Netherlands. Her competencies include Biochemistry and Green & Sustainable Chemistry, and she has made significant contributions to Proteomics and Molecular Enzymology. She has an impressive research output, with numerous articles and conference contributions. Her work contributes towards the global Sustainable Development Goals (SDGs) to end poverty, protect the planet, and ensure prosperity for all. She is currently the Vice-Rector for Research, Innovation, and Community Development at Universitas Airlangga and the Head of the Research Center for Bio-Molecule Engineering at the same university. During the COVID-19 pandemic, Prof. Ni Nyoman coordinated the COVID-19 research product at UNAIR. You can contact Prof. Dr. Ni Nyoman Tri Puspaningsih via email at ni-nyoman-t-p@fst.unair.ac.id.