International Journal of Informatics and Computation (IJICOM)
Vol. 4, No.1, August 2022
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

# Robust Prediction Model of Covid-19 using Deep Learning algorithm

Rike Pradila

**Abstract**
Predicting the Covid-19 outbreak in a large dataset is a difficult task and complex problem. Many communities have put up different approaches to forecast Covid-19 positive cases. Conventional methods continue to have problems predicting the real trend situations, nevertheless. In this experiment, we use CNN to develop our model by examining attributes from the enormous Covid-19 dataset to anticipate long-lasting outbreaks and provide early intervention. Based on the outcomes of the experiment, our model can achieve sufficient accuracy with a negligible loss. In this study, we compute the function that yields RMSE 0.00070 and MAPE 0,02440 for new case prediction and RMSE 0.00468 and MAPE 0.06446 for new death prediction. As a result, our suggested strategy can accurately forecast the trend of positive cases in Indonesia.
***Keywords:***
Covid-19, Prediction, Deep Learning, CNN



## 1. Introduction

The WHO claims that Covid-19 was a global epidemic of international concern because of how quickly it has spread to various nations [1][2][3]. Because of Covid-19's ease of dissemination, countries have implemented measures to stop the virus from spreading, including travel bans, quarantines, event cancellations or postponements, social isolation, and lockdowns [4]. This problem exacerbates issues with global public health that have negative effects on things besides human health, like mental stress, a volatile economy, and other detrimental daily activities. As a result, governments around the world are attempting to decide on public policy for the economy and for health [5], [6].

The Covid-19 problem must be handled, however using traditional methods to predict cases is difficult [7]. A precise plan for control methods can be made by forecasting and researching the pattern of disease spread. Using forecasting algorithms-based predictive models, they create the prognosis. Every day sees a different rise in Covid-19 positive cases, making it challenging to control the spread. In this situation, DL can offer improved forecasts and advancements [8]. In order to provide early warning and determine how activities would affect the transmission of the virus, intelligence methods are crucial in anticipating the development of Covid-19. In order to predict and assess trend cases for Covid-19, multiple intelligence techniques have been used in the present model development [1].

An RNN to estimate confirmed cases with LSTM was proposed in a study [9] [10] [11]. According to several reports, Deep LSTM, Convolutional LSTM, and Bi-LSTM can be used to predict the augmentation of positive instances on a daily and weekly basis [4]. With the LSTM, GRU, and Bi-LSTM learning models, other research have also been able to predict

**Corresponding Author:**
Rike Pradila, Universitas Respati Yogyakarta, 18220014@respati.ac.id

Covid-19 [1]. When used with a random test set, the approach can yield findings that are more useful than ANN [13]. Another study investigated the use of ANN and mathematical and computational models to forecast confirmed instances [7].

A technique based on contemplative fuzzy logic to examine conflicting parameters is presented in various studies to anticipate the spread [14]. ANFIS and LSTM prediction models were utilized in a different investigation to forecast newly infected cases. Input gates, forget gates, control gates, and output gates are the four gates that make up the LSTM architecture. The first layer builds the second layer, the last layer normalizes the function and sends it to the last layer, which is the output layer, while ANFIS uses the input layer to receive parameters. ANFIS has more layers than LSTM, hence it needs more hardware and training time [8].

Another study calculated ARIMA to examine spatial as a short term to measure confirmed cases. The ARIMA model is capable of acquiring AR for taking into account past values and MA for assessing both the present and prior residual series historical knowledge [15]. SVR, NN, and Linear Regression are the three regression methods used by the ARIMA approach [16] [17]. The accuracy of the ARIMA model in predicting future Covid-19 cases demonstrates the value of epidemiological surveillance. Events with significantly higher intervals can be predicted using the ARIMA approach [18].

It is now difficult to measure the spread trend using Covid-19 prediction. By analysis and prediction of new cases and death cases using the DL approach, the current paper proposed positive cases for Covid-19 utilizing CNN. CNN can also estimate precise outcomes in time series analysis since it can examine the dataset's properties [19]. In order to create algorithms using RMSE and MAPE indicators and evaluate the model, the paper used CNN. The model can more accurately anticipate Covid-19 trend cases based on the experiment's findings [5] [20].

Therefore, in this study, we provide a method for accurately representing the new cases and new death of Covid-19 in a prediction model built using CNN. We provide numerous important advances to Covid-19 prediction research, particularly in case categorization utilizing the following learning techniques:

1. We using CNN to train datasets for useful models and we present a new method for anticipating new cases and new deaths. As part of the feature datasets, we used to develop our learning model, we included the date, recent cases, and death cases.
2. We create a model to forecast Covid-19 positive cases. Finding the cases instantly might be possible with the help of our prediction model. Also, the suggested method may be a potential way to enhance the existing model and generate a higher score.
3. We test the suggested model to obtain very accurate findings to swiftly and reliably predict new cases and new deaths in the upcoming few days. In order to acquire the best results, we train broad features and adjust the settings to reach the best accuracy numbers.

The structure of this document is as follows: Part II dives deeper into earlier discoveries. The issue description is discussed in Section III. Although Part V presents the results and in-depth analysis, Part IV describes the experimental design, including a feature learning algorithm, a dataset, and preprocessing. The research's unanswered issues are compiled in Part VI.

## 2. Related Works

Research into deep learning is expanding to address a variety of problems. [21][22][23][24]. An article in the Covid-19 prediction research suggested ANFIS as a fuzzy logic-based variation of ANN. In a different study, ANFIS and LSTM give excellent results when dealing with non-linear data using computational methods to estimate new Covid-19 infection cases. ANFIS and LSTM, however, are only able to predict and cannot generate meaningful hypotheses about this outbreak. This explains why it is difficult to demonstrate accuracy, especially for short datasets, and why the Covid-19 prediction model cannot prevent failure. According to the experimental findings, LSTM outperforms ANFIS [8].

Another study predicts Covid-19 results using mediative fuzzy. The increases of positive patients and the passing of time in terms of the increment are related by the meditative fuzzy correlation approach. In statistical analysis, the correlation coefficient is crucial in identifying the linear relationship between two independent variables. The correlation coefficient of contemplative fuzzy logic, however, examines numerous opposing parameters. In order to do this, the first lower and upper bounds of the fuzzy correlation were used to construct the mediative fuzzy correlation coefficient. The correlations found from the calculations of fuzzy logic, intuitive fuzzy logic, and mediative fuzzy logic are [0.25, 0.37], [0.2294, 0.39], and [0.21, 4092] [14].

In order to address the Covid-19 problem, multiple articles presented deep learning variations (RNN) that could accurately forecast daily and weekly cases. The method used in the article yields excellent short-term prediction accuracy, with an error for daily forecasts of less than 3% and for weekly forecasts of less than 8%. [4]. Another article evaluated the influence of preventive measures like social isolation and lockout on the spread of Covid-19 as well as anticipating the number of cases in a month using data-driven estimating methodologies like LSTM and curve fitting. The quantity of recovered instances and the quantity of positive cases were both utilized to forecast specific variables. [9].

To locate confirmed deaths and recovered cases, other articles suggested using LSTM, GRU, and Bi-LSTM. To get around RNN's constraints, LSTM can also use memory cells, which are hidden layer units. The memory cell's link allows it to store the temporal state and control through the gate. Moreover, a simple model lacks a GRU, a type of memory cell used to store data. As a result, the GRU is limited to controlling the data inside the unit. Bi-LSTM can improve memory retention and learning capacity. The DL technique utilizing Bi-LSTM in particular results in reduced mistakes and greater precision. LSTM, GRU, and Bi-LSTM work together to produce MAE and RMSE of 0.0070 and 0.0077 [1].

Using various training techniques, a study investigated the use of ANN to predict Covid-19 fatalities. Levenberg-Marquardt and resilient propagation are the technique. By combining velocity and convergence, the Levenberg-Marquardt approach can be used to solve numerical issues for non-linear functions. In addition, using the local gradient data from the weight step, the Resilient Propagation method can directly adapt. The MSE produced by the Resilient Propagation function is ten times higher than the MSE produced by the Levenberg-Marquardt approach. As a result, ANN with a random set can outperform ANN with a specific group [13].

Another study used Gompertz and Logic as a mathematical approach and ANN as a computational approach to predict Covid-19 instances. The three-layer ANN model has an input layer that assigns weights to each parameter. The transfer function is used by the hidden layer to characterize the internal output. As a result, the hidden layer's added coefficient and the output layer's simulated signal weight are both produced. To compare the differences between the cases that were observed and the situations that each model anticipated, several parameter models apply analytical methods in mathematical models. It generates R2, which is represented by the Gompertz model of 0.9998, Logistics of

0.9996, and ANN of 0.9999, based on the estimated data and observed in computerized methods and mathematical models employing all confirmed examples. The Gompertz and Logistic models are outclassed by these two ANN models [7].

A number of communities also suggested using ARIMA for spatial distribution analysis. Efficiency measures like the enhanced yield index, MAE, and RMSE, however, are not appropriate for precise model prediction. Nonetheless, the ARIMA model's success in predicting future Covid-19 epidemics demonstrates its efficacy in epidemiological surveillance [18]. Several studies employed estimate techniques for the Covid-19 dataset and ARIMA to collect data. In this study, ARIMA is used to get a straightforward average number depending on the effectiveness of the regression method. Nevertheless, using the ARIMA method only yields the best regression, RMSE 286,879, MSE 78604,436, and MAE 175,672 [16], without increasing the matrix error. Nonetheless, the ARIMA model is capable of forecasting upcoming Covid-19 infections. ARIMA is not recommended for non-linear connections, particularly major and dynamic complexity issues, according to certain academics. Due to its dependence on the finite model, ARIMA is unable to capture hidden non-linear time series. The Parson correlation coefficient is used in this experiment to generate relative confidence of 95%, while the actual point estimate data is 0.996 [15].

Based on CNN, the recent publications presented a novel method for anticipating future cases and fatalities. Daily fresh instances and deaths were collected for the study's dataset. The experiment also demonstrates that early CNN input characteristics and later inputs greatly affect CNN structure, aiding in case prediction. In comparison to previous DL techniques, the suggested CNN model can achieve the highest predicted efficacy and accuracy. Several studies demonstrate that CNN outperforms its tested competitors [5].

A different study suggested CNN as a better way to forecast positive cases than models like LSTM, GRU, and MCNN. The CNN model beats other Models in validation accuracy and forecasting consistency, according to the analysis's findings. The proposed CNN model's research of essential features, invariant distortion, and temporal dependency learning can have powerful long-term effects in time series analysis. Because of its deep feature learning capacity, CNN is the best forecasting model [19].

In order to address those problems, we suggest a model to forecast future cases by adding a novel CNN method based on dataset attributes. In order to create our dataset and train our model, we gather a sizable Covid-19 case. This model is able to forecast instances of Covid-19 issues in the coming days by training a number of informative features.

# 3. Proposed Method

A formal statement of the study problem and a few of the journal's concepts will be given in this part.

A. Problem Definitions

Input, hidden, and output are the three layers that make up CNN [25]. Conventional CNNs project the extraction as an input matrix using the input layer. In the dataset, there are features $x\ (x1, x2, x3, \dots\ xn)$, $s\ \rightarrow\ Training\ samples$, $c\ \rightarrow\ Batch\ size\ a\ \rightarrow\ 0$ for $i$ to epoch then for to $s$ then $j\ =\ k\ mod\ c$

$$if\ j\ =\ 0\ then \quad j\ =\ c \quad end \tag{1}$$

$$if\ j\ -\ a\ >\ 0\ then \tag{2}$$

Using equations (1) and (2), the training sample set to be output in batches.

else

Equations (1) and (2) use the loss function to calculate the loss, the optimization to update the weights, and the training set sample to calculate the output in the subsequent batch.

end

$a = j\, end$

$end\ Testing$:

The CNN model predicts the testing set, evaluates the difference between the actual and predicted values, and calculates the prediction error.

B. Proposed Method

In this study, a DNN with numerous hidden layers was utilized to train and test a Covid-19 case prediction model using CNN. Also, we experiment with gradient descent when using objective function parameter models. To optimize and shorten training time, we adopt a pooling layer on the neural network instead of the customary pooling layer. In order to affect neural units, CNN combines the two layers, including kernel size and pooling.

To get the best results with the varied input vector, we compute the losses from the training and testing process using the CNN model. This work computes a dataset with the appropriate hyperparameter values using a 1D dataset time series [26]. By defining calculation over NN as follows, we create a supervised learning model: Equation (1) the objective function of an image in Faster RCNN is given as follows:

Input features $x^{(i)} \in R$

Outputs $x^{(i)} \in Y\ (e.g. R, \{0,1\}, \{1, \ldots., p\})$

Model Parameters $\theta \in \mathbb{R}^k$

Hypothesis function $h_{\theta:} : \mathbb{R}^n \rightarrow \mathbb{R}$

Loss function $\ell : \mathbb{R} \times Y \rightarrow \mathbb{R}_{+:}$

Equation (3) in this work, the optimization problem was calculated as follows:

$$\underset{\theta}{Minimise} \sum_{i=1}^{m} \ell\ (\mathrm{h}\theta : (x^{(i)}), y^{(i)}) \quad (3)$$

In this paper, we present the neural network processing hypothesis function $h_\theta : \mathbb{R}^n \rightarrow \mathbb{R}$. We must compute the forward pass and the backward pass on a CNN in order to get the gradient of the loss function in the model. Equation (4) (4) In order to obtain convolution output $z_{i:}$, the study calculates the forward pass to convolve input matrix $x_i$ with filter $W_i$ as follows:

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (4)$$
$$z_{i:}(x_i) = W_i x_i + b$$

The parameters of the convolutional layer of the CNN during training are the filters $W_i$ and bias term $b$. The input, representation, and metrics necessary to compute tensors in the hidden layer are all provided by this supervised learning model. In order to conduct complex model computations with a small number of parameters, CNN includes many similar neurons spread throughout its layers. The feature maps serve as the layer's only input, and the feature maps are computed as its output by convolutioning filters across the feature maps. During training, learning is accomplished using the back-propagation model and filter parameters of the convolution layer. Equation (5) From the Jacobian matrix $m\ x\ n$, we compute the vector-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ in the backward pass.

$$\left(\frac{\partial f(x)}{\partial x}\right) \in \mathbb{R}^{m\,x\,n} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \frac{\partial f_m(x)}{\partial x_2} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix} \quad (5)$$

With CNN's layer pooling method, statistics from the closest output at a particular point can be output. The representation is made more resilient and nearly invariant to small input changes thanks to the pooling layer. The pooling layer also lessens the quantity of intermediate representations, which lowers the ability to provide Covid-19 predictions [27].

# 4. Experimental Setup

A. Main Idea

The main goal of this study is to use the CNN algorithm to build a prediction model based on daily new cases and new deaths. During the training process, the CNN computes the most informative features using a number of hidden layers and evaluates the model's performance [9]. Because to its feature studies, CNN can have a significant long-term impact on time series analysis [19]. As a result, many communities suggested CNN create a model to forecast the Covid-19 trend. This technique's goal is to classify the data into pre-existing groups. We use the design because it is highly accurate and successful in solving problems involving long-term forecasting [28] [29].

B. Dataset

In this study, we gather the Covid-19 dataset, which includes entity, code, day, new cases, and new death. A time-series dataset from March 2, 2020 to November 16, 2021 is included in this experiment. Then, we separate it into testing datasets to assess the model's performance and training datasets to build the model. The 625 data with five features in the whole dataset are divided in this experiment by 80% for training and 20% for testing. The distribution of the study's datasets is shown in detail in table 1 as follows:

Table 1. Distribution of the dataset

| Dataset | Sample |
|---|---|
| Data Training (80%) | 500 |
| Data Validation (20%) | 125 |
| **Total** | **625** |

C. Data Preprocessing

In data preprocessing, we transform raw datasets into information by cleaning, filtering, and combining data. After data processing, we fill up the gaps left by missing values by removing irrelevant information. After filling in the missing value, we divided the dataset into multiple features to examine the data types of each variable and check for empty or NA values [30]. After that, remove the noisy data to make the data useful [31]. After that, complete the processes of vectorization, normalization, and feature extraction [32].

D. Prediction Method

In this work, CNN is used to create a prediction model to examine the Covid-19 case's trend. To carry out our research, we gather datasets such as entity, code, day, new cases, and new death. We separate the dataset into training and testing datasets after it has been gathered. The second stage is data preprocessing, where we adjust the format of the raw data and clear the missing value procedure to create samples that are appropriate. Then, in order to check for empty data, we reduce the dataset's noise for each variable. By removing the noisy data, we may acquire a useful dataset for the training and testing procedure. We use the training dataset to train the model after the preprocessing phase. In order to assess or gauge the model performance and test our model, this study uses a testing dataset. We offer unknown datasets during the testing phase to obtain the ideal model, and we adjust some hyperparameters to obtain the optimum accuracy value. Finally, the suggested approach can generate a useful model for anticipating COVID-19 positive cases [33].

# 5. Result & Analysis

A. Prediction Test

In this work, we use the learning model to build our model by gathering the most useful feature for predicting Covid-19 cases to estimate the number of difficulties. The study collects the dataset made up of fresh positive cases and fresh death characteristics. In order to forecast cases of Covid-19 during the next 10 days, this experiment examines a 30-day model. Figures 1 and 2 illustrate graphs of the prediction of new cases and deaths, respectively, in the study.

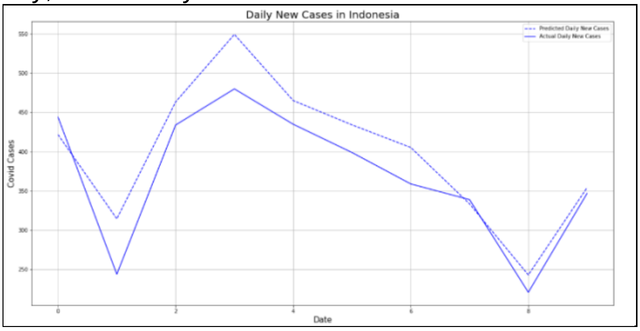


Fig. 1 Prediction graph for new cases in the next ten days.

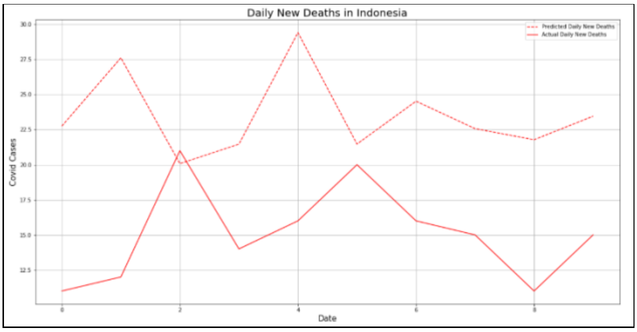


Fig. 2 Prediction graph for new deaths in the next ten days.

The dashed line plot in Fig. 1 displays the predictions for new cases, and the blue line plot displays the actual daily new Covid-19 instances. On a range of 0 to 550 new cases over the course of the next ten days, the daily new case graph offers predictions that are fairly accurate for Covid-19. A dashed line plot and a red line plot depicting actual daily cases of Covid-19 deaths, respectively, in Fig. 2's prediction results for death cases. On a scale of 0 to 30 new deaths in the next ten days, the daily new death graph generates predictions that are reasonably close to the actual new death of Covid-19.

Label Y is the daily average of new cases and deaths, and label X is the date for projecting ten days of Covid-19 instances based on the graph above. Plot the dotted line to compare the case's prognosis to the actual circumstance. This demonstrates how close to the actual daily cases the prediction algorithm can get a result. Based on the results of the experiment, our suggested model using CNN can successfully predict a result that is close to the actual case line pattern. Following numerous training stages, the prediction technique can increase accuracy in the Covid-19 prediction situations with a negligible loss.

B. Evaluation Metric

Predictive models use a range of assessment criteria to gauge accuracy. To test the prediction model, we computed RMSE and MAPE on the predicted outcomes. As a result, lower error rates and smaller RMSE and MAPE values indicate more accurate results. Equation (6) (6) The following formula is used to determine the RMSE of Dan Equation (7) MAPE:

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_l)^2} \tag{6}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_l}{y_i}\right| \times 100\% \tag{7}$$

In this equation, $y_i$ stands for the actual value, $\hat{y}_l$ for the anticipated value, and $n$ for the number of data points.

Table 2 Results of RMSE and MAPE scores

| | RMSE | MAPE |
|---|---|---|
| New Cases | 0.00070 | 0.02440 |
| New Death | 0.00468 | 0.06446 |

The results of the parameters used to calculate the error value using epochs = 1000 are shown in Table 2. The CNN algorithm-based Cases Covid-19 forecasts reveal a reasonably low error value, with new cases having an RMSE of 0.00082, MAPE of 0.02440, and new deaths having an RMSE of 0,00468, MAPE of 0.06446. These findings demonstrate that our suggested model can provide less RMSE and MAPE values, demonstrating a superior prediction capability.

# 6. Conclusion

Several researches employ traditional methods to foretell the positive instances in the spread of Covid-19. Conventional approaches still have problems reliably forecasting the Covid-19 trend situations, though. In order to accurately anticipate Covid-19 instances in Indonesia, we build a unique learning model utilizing CNN to address this issue. To build our prediction model in this work, we collect a daily Covid-19 case as our dataset. To forecast the trend using various hidden layer parameters, we use the trend cases dataset. The study adjusted a number of hyperparameters to provide results with excellent accuracy.

By using several regulators and long epochs, we tune numerous different hyperparameters during testing to obtain high accuracy and minimal loss. The model can achieve accuracy in new cases with RMSE 0.00070, MAPE 0.02440, and RMSE 0.00468, MAPE 0.06446 for new deaths based on the experimental results. In addition, this study does more than just measure accuracy and loss; we also assess the model's performance. As a result, the suggested approach may offer a potential means of addressing the Covid-19 prediction issue in the present.

Future study will involve adding a different algorithm to this model using the GCN architecture in order to improve the prediction outcome. GCN is a CNN extension that can function directly on a graph. With an additional hyperparameter setting, the architecture GCN can be a creative way to create higher accuracy.

## Acknowledgment

This paper is conducted in the Department of Informatics, Respati University of Yogyakarta, Indonesia.

## References

[1] F. Shahid, A. Zameer, and M. Muneeb, “Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM,” Chaos, Solitons and Fractals, vol. 140, p. 110212, 2020.
[2] A. Senapati, A. Nag, A. Mondal, and S. Maji, “A novel framework for COVID-19 case prediction through piecewise regression in India,” Int. J. Inf. Technol., vol. 13, no. 1, pp. 41–48, 2021.
[3] J. Song, H. Xie, B. Gao, Y. Zhong, C. Gu, and K. S. Choi, “Maximum likelihood-based extended Kalman filter for COVID-19 prediction,” Chaos, Solitons and Fractals, vol. 146, p. 110922, 2021.
[4] P. Arora, H. Kumar, and B. K. Panigrahi, “Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India,” Chaos, Solitons and Fractals, vol. 139, p. 110017, 2020.
[5] C. J. Huang and P. H. Kuo, “Multiple-Input Deep Convolutional Neural Network Model for Short-Term Photovoltaic Power Forecasting,” IEEE Access, vol. 7, pp. 74822–74834, 2019.
[6] V. Singh, R. C. Poonia, S. Kumar, P. Dass, and P. Agarwal, “Prediction of COVID-19 corona virus pandemic based on time series data using support vector machine,” J. Discret. Math. Sci. Cryptogr., vol. 23, no. 8, pp. 1583–1597, 2020.
[7] O. Torrealba-Rodriguez, R. A. Conde-Gutiérrez, and A. L. Hernández-Javier, “Modeling and prediction of COVID-19 in Mexico applying mathematical and computational models,” Chaos, Solitons and Fractals, vol. 138, 2020.
[8] A. A. Chowdhury, K. T. Hasan, and K. K. S. Hoque, “Analysis and Prediction of COVID-19 Pandemic in Bangladesh by Using ANFIS and LSTM Network,” Cognit. Comput., vol. 13, no. 3, pp. 761–770, 2021.
[9] A. Tomar and N. Gupta, “Prediction for the spread of COVID-19 in India and effectiveness of preventive measures,” Sci. Total Environ., vol. 728, p. 138762, 2020.
[10] V. K. R. Chimmula and L. Zhang, “Time series forecasting of COVID-19 transmission in Canada using LSTM networks,” Chaos, Solitons and Fractals, vol. 135, 2020.
[11] J. Luo, Z. Zhang, Y. Fu, and F. Rao, “Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms,” Results Phys., vol. 27, p. 104462, 2021.
[12] N. Ayoobi et al., “Time series forecasting of new cases and new deaths rate for COVID-19 using deep learning methods,” Results Phys., vol. 27, no. June, p. 104495, 2021.
[13] L. S. de Oliveira, S. B. Gruetzmacher, and J. P. Teixeira, “Covid-19 time series prediction,” Procedia Comput. Sci., vol. 181, no. 2019, pp. 973–980, 2021.
[14] M. K. Sharma, N. Dhiman, Vandana, and V. N. Mishra, “Mediative fuzzy logic mathematical model: A contradictory management prediction in COVID-19 pandemic,” Appl. Soft Comput., vol. 105, p. 107285, 2021.
[15] H. Alabdulrazzaq, M. N. Alenezi, Y. Rawajfih, B. A. Alghannam, and A. A. Al-Hassan, “On the accuracy of ARIMA based prediction of COVID-19 spread,” Results Phys., vol. 27, p. 104509, Aug. 2021.
[16] R. O. Ogundokun and J. B. Awotunde, “Machine learning prediction for COVID 19 pandemic in India,” medRxiv, vol. 2019, 2020.
[17] F. Rustam et al., “COVID-19 Future Forecasting Using Supervised Machine Learning Models,”

IEEE Access, vol. 8, pp. 101489–101499, 2020.
[18] S. Roy, G. S. Bhunia, and P. K. Shit, “Spatial prediction of COVID-19 epidemic using ARIMA techniques in India,” Model. Earth Syst. Environ., vol. 7, no. 2, pp. 1385–1391, 2021.
[19] K. N. Nabi, M. T. Tahmid, A. Rafi, M. E. Kader, and M. A. Haider, “Forecasting COVID-19 cases: A comparative analysis between recurrent and convolutional neural networks,” Results Phys., vol. 24, p. 104137, 2021.
[20] S. Dutta, S. K. Bandyopadhyay, and T. H. Kim, “CNN-LSTM Model for Verifying Predictions of Covid-19 Cases,” Asian J. Res. Comput. Sci., vol. 5, no. 4, pp. 25–32, 2020.
[21] M. Diqi, U. Universitas, R. Yogakarta, N. Wijaya, U. Universitas, and R. Yogakarta, “Pooling Comparison in CNN Architecture for Javanese Script Classification,” no. January, 2022, doi: 10.35842/ijicom.
[22] H. J. Jie and P. Wanda, “Runpool: A dynamic pooling layer for convolution neural network,” Int. J. Comput. Intell. Syst., vol. 13, no. 1, pp. 66–76, 2020, doi: 10.2991/ijcis.d.200120.002.
[23] P. Wanda, Selo, and B. S. Hantono, “Model of secure P2P mobile instant messaging based on virtual network,” 2014 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2014 - Proc., no. November, pp. 81–85, 2014, doi: 10.1109/ICITSI.2014.7048242.
[24] P. Wanda and H. Jin Jie, “URLDeep: Continuous Prediction of Malicious URL with Dynamic Deep Learning in Social Networks,” Int. J. Netw. Secur., vol. 21, no. 6, pp. 971–978, 2019, doi: 10.6633/IJNS.846.
[25] C. Prayadi, M. Diqi, and S. H. Mulyani, “Implementation of Cnn for Plant Leaf,” Int. J. Informatics Comput., vol. 2, no. 2, 2020.
[26] P. Wanda, M. E. Hiswati, and H. J. Jie, “DeepOSN: Bringing deep learning as malicious detection scheme in online social network,” IAES Int. J. Artif. Intell., vol. 9, no. 1, pp. 146–154, 2020.
[27] C. Pan, M. Lu, B. Xu, and H. Gao, “An improved CNN model for within-project software defect prediction,” Appl. Sci., vol. 9, no. 10, pp. 1–28, 2019.
[28] R. G. Babukarthik, V. A. K. Adiga, G. Sambasivam, and D. Chandramohan, “Prediction of covid-19 using genetic deep learning convolutional neural network (GDCNN),” IEEE Access, vol. 8, pp. 177647–177666, 2020.
[29] J. Frausto-Solís, L. J. Hernández-González, J. J. G.-B. J. P. Sánchez-Hernández, and E. Román-Rangel, “Convolutional Neural Network–Component Transformation (CNN–CT) for Confirmed COVID-19 Cases,” Math. Comput. Appl., vol. 26, no. 2, p. 29, 2021.
[30] A. U. Mandayam, Rakshith. A. C, S. S, and S. K. Niranjan, “Prediction of Covid-19 pandemic based on Regression,” Proc. - 2020 5th Int. Conf. Res. Comput. Intell. Commun. Networks, ICRCICN 2020, pp. 1–5, 2020.
[31] R. M. A. Latif, S. B. Belhaouari, S. Saeed, L. B. Imran, M. Sadiq, and M. Farhan, “Integration of Google Play Content and Frost Prediction Using CNN: Scalable IoT Framework for Big Data,” IEEE Access, vol. 8, pp. 6890–6900, 2020.
[32] M. Diqi and A. David, “Effective Soil Type Classification Using Convolutional Neural Network,” Int. J. Informatics Comput., vol. 3, no. 1, 2021.
[33] A. Abidin, Hamzah, and M. E. H, “Efficient Fruits Classification Using Convolutional Neural Network,” Int. J. Informatics Comput., vol. 3, no. 1, 2021.