Journal of Applied Data Sciences
Vol. 5, No. 4, December 2024, pp. 1782-1789

Enforcement of Community Activity Restrictions Level Prediction in Jakarta Using Long Short-Term Memory Network

Chendra Dewangga[1], Seng Hansun[2, *]

[1,2] *Informatics Department, Universitas Multimedia Nusantara, Jl. Scientia Boulevard, Gading Serpong, Banten, 15810, Indonesia*

(Received: August 19, 2024; Revised: September 12, 2024; Accepted: October 01, 2024; Available online: October 15, 2024)

Abstract

The implementation of restrictions on community activities (Pemberlakuan Pembatasan Kegiatan Masyarakat – PPKM) is a strategy from the Indonesian government in handling the spread of COVID-19. PPKM is divided into four levels which will determine the restriction types that are to be implemented in a region. In this study, we aim to build a website that can predict PPKM levels through COVID-19 daily positive and death cases recorded in the Jakarta City, Indonesia. The prediction system uses the Long Short-Term Memory (LSTM) network and Node.JS as the backend of the website. We also introduced the usage of multivariate approach for this regression task by combining both daily positive and death cases number into the LSTM network. Based on the test scores obtained through evaluation using Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE), it was concluded that the proposed LSTM method could accurately predict the death cases with 0.17% MAPE and 22.68 RMSE but has poor performance in predicting the daily positive cases with 53.11% MAPE and 27.15 RMSE. This might be rooted from the use of multivariate approach during the model development where more variation to the daily positive cases was detected.

Keywords: COVID-19, Machine Learning, Prediction, PPKM, LSTM

1. Introduction

Since the World Health Organization (WHO) has declared that the Coronavirus Disease 2019 (COVID-19) as a pandemic in March 2020 [1], the President of the Republic of Indonesia and the Regional Government have worked together to take several steps to prevent the spread of the COVID-19 in the community [2]. COVID-19 or also known as Sars-CoV-2 is a pandemic virus that first spread in China and is a zoonotic virus (transmitted between humans and animals). It can easily spread via droplets (small particles) when someone talks or sneezes, air, and contaminated surfaces [3].

The COVID-19 pandemic has various negative impacts on the community, such as reduced public consumption, many production activities will stop, and lead to a situation of uncertainty [4]. One of the government's strategic efforts to deal with COVID-19 in Indonesia is by establishing regulations for the Enforcement of Community Activity Restrictions [5]. PPKM is divided into four levels (level one, two, three and four) and each level affects activities in the community.

Previously, there was a study conducted which analyzes the effect of PSBB (pembatasan sosial berskala besar – predecessor of PPKM policy) and PPKM policies on the decline in COVID-19 cases in Indonesia using the Google Mobility Index and their impact on the economy in Indonesia [6]. The conclusion obtained is that policies to reduce community mobility in reducing the risk of the spread of COVID-19 are prioritized in line with the delta variant which spreads faster than the previous variant. In economic recovery, the government can provide capital injections in retail and recreational areas, grocery stores, food and pharmacies, especially after the spread of COVID-19 can be controlled [6].

Another similar research has been carried out to predict the spread of COVID-19 in Jakarta with several time series models, including the Holt's exponential smoothing and Auto-Regressive Integrated Moving Average (ARIMA) [7]. The study used both models to forecast the number of COVID-19 cases in Jakarta between March 1 and July 6, 2020.

***Corresponding author: Seng Hansun (seng.hansun@lecturer.umn.ac.id)**
DOI: https://doi.org/10.47738/jads.v5i3.318

From the experimental results, ARIMA achieved the highest R-squared and lowest Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) confirming the capability of the model to fit and predict the upcoming number of infected COVID-19 cases in Jakarta [7].

Because the percentage error rate is still higher than the standard, the prediction of changes in the PPKM level in the Jakarta area can be done using another, more advanced method called the Long Short-Term Memory (LSTM). LSTM is a popular deep learning method in time series domain and the performance tends to be more accurate than ARIMA [8]. The LSTM method itself has a higher level of complexity than ARIMA because the LSTM cell consists of a combination of several gates in regulating the calculation process.

In an effort to assist the government in managing the transmission of COVID-19 in the Jakarta City area, it is necessary to have a system that can predict the number of daily cases and daily deaths due to COVID-19 in Jakarta and provide recommendations for implementing PPKM levels that are considered appropriate. This can also motivate research on strategic actions to be taken when new unforeseen pandemic happens in the future.

2. Method

2.1. Long Short-Term Memory

Long Short-Term Memory (LSTM) is a derivative of the Recurrent Neural Network (RNN) which has been proven to be successful in predicting time series data [9]. LSTM generally consists of three components that can regulate the flow of information, namely forget gate, output gate, and input gate [10]. The LSTM, initiated by Hochreiter and Schmidhuber, is an improved version of the RNN where the LSTM is equipped with a memory cell that can store information for a long time [11]. Compared to LSTM, RNN cannot hold long memory, so the use of LSTM based on "memory line" proves to be very useful in case forecasting with a large amount of data [12]. In one LSTM cell, there are four components involved [13], namely:

Forget Gate (f_t) that has a function to determine how much information from the new input (x_t) and previous hidden state value (h_{t-1}) will be forwarded to the current cell state. The formulation is shown in Eq.(1).

$$\mathrm{f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)} \tag{1}$$

Input Gate ($\mathrm{i_t}$) that has a function to determine which part of the data will be updated. It involves two formulations as shown in Eq.(2) to calculate the output from the input gate and Eq.(3) to determine the candidate cell state value ($\tilde{\mathrm{C}}_\mathrm{t}$).

$$\mathrm{i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)} \tag{2}$$

$$\tilde{\mathrm{C}}_\mathrm{t} = \mathrm{tanh(W_C h_{t-1} + U_C x_t + b_C)} \tag{3}$$

Cell State ($\mathrm{C_t}$) which is used to update input $\mathrm{C_{t-1}}$ (old value) with new input value. The formulation is shown in Eq.(4).

$$\mathrm{C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t} \tag{4}$$

Output Gate (o_t) consist of two parts, first to calculate the output value as shown in Eq.(5) and second to calculate the hidden state value (h_t) within the memory cell as shown in Eq.(6).

$$\mathrm{o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)} \tag{5}$$

$$\mathrm{h_t = o_t \odot tanh\ (C_t)} \tag{6}$$

Here, W_f, W_i, W_C, W_o and U_f, U_i, U_C, U_o are weights and b_f, b_i, b_C, b_o are bias values.

An LSTM cell structure can be illustrated as shown in figure 1. LSTM in this study will use data on positive and death cases due to COVID-19 with daily time intervals. The data will be normalized first, then training will be carried out to get the LSTM prediction model. After that, the model will be used to make predictions and errors are calculated using the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE).

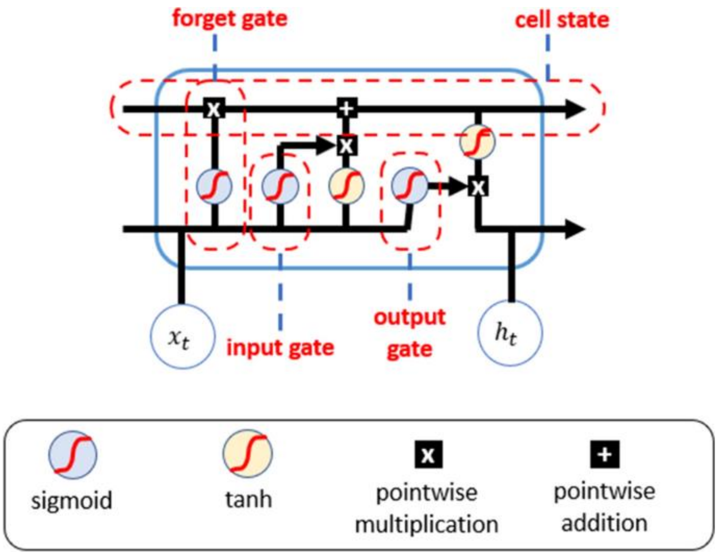

Figure 1. An LSTM cell [10]

2.2. Root Mean Square Error and Mean Absolute Percentage Error

RMSE is the standard deviation of the prediction error or residual, while MAPE is a method used to measure the accuracy of forecasting methods. MAPE represents the average absolute percentage error of each entry in the data set, indicating on average, how accurately the predicted number (F_t) is compared to the actual number (Y_t). Both of them can be represented in Eq.(7) and Eq.(8) [14], [15].

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_t - F_t)^2} \tag{7}$$

$$\mathrm{MAPE} = \left(\frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - F_t}{Y_t}\right|\right) \cdot 100\% \tag{8}$$

2.3. Research Methodology

Figure 2 shows the research methodology in this study. The dataset used is starting from January 1st, 2021 to January 1st, 2022 and can be downloaded directly through the official government website. Data that has been filtered according to the required date and column will be immediately normalized. The purpose of data normalization is to scale the attribute values within a certain range [16]. In this study, we use MinMaxScaler starting from a scale of 0 to 1.

After that, the new data is split into two parts, about 80% of the data is used for training the model and the other 20% is used for testing the model. Then the data is divided into two parts, namely into x train and y train (x is the input variable and y are the output variable to be achieved).

The network architecture consists of two LSTM layers and one dropout layer which will be entered into a function like as shown in figure 3. We started by calling the Sequential function to define a structural deep learning architecture. Next, we created an LSTM layer, followed by a dropout layer to manage overfitting, and another LSTM layer. In the last output layer, we added a Dense layer with two neurons.

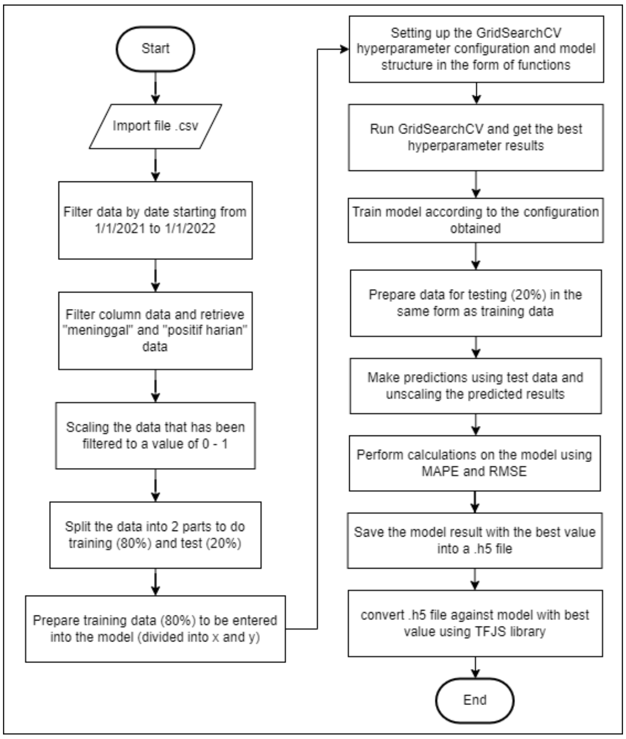

Figure 2. Research methodology

```
def define_model_grid(dropout_rate, first_neuron, second_neuron):
    model_gscv = Sequential()
    model_gscv.add(LSTM(first_neuron, input_shape=(x_train.shape[1], x_train.shape[2]), return_sequences=True))
    model_gscv.add(Dropout(dropout_rate))
    model_gscv.add(LSTM(second_neuron, return_sequences=False))
    model_gscv.add(Dense(y_train.shape[1]))
    model_gscv.compile(optimizer='adam', loss='mean_squared_error', metrics=['acc'])

    return model_gscv
```

Figure 3. The structure of the network

In addition, there is a hyperparameter configuration that has been determined with the aim of being included in the GridSearchCV function which basically calculates all possible combinations to find the best parameter. The configuration can be seen in figure 4.

```
dropout_rate = [0.2, 0.3, 0.4]
first_neuron = [128, 64, 32]
second_neuron = [128, 64, 32]
batch_size = [10, 20, 30]
epoch = [20, 25]
```

Figure 4. Hyperparameter configuration

The data to be observed is multivariate which means that the observed data is more than one data, in this case, daily positive and deaths cases. An illustration of the model made can be seen in figure 5. The purpose of using multivariate in this prediction is to make prediction time more efficient.

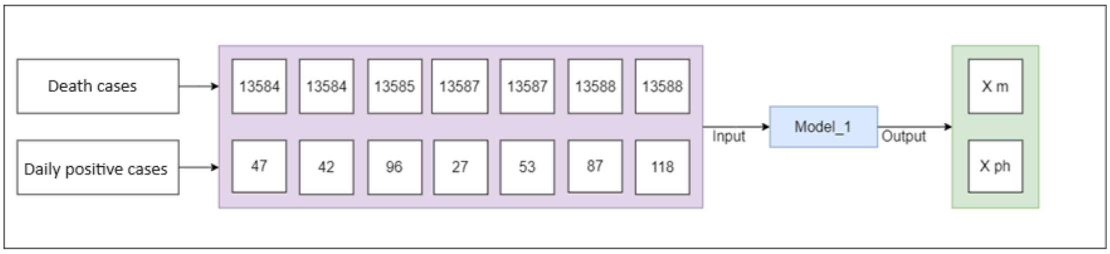

Figure 5. Illustration of the multivariate model prediction process

After the data has been normalized and the model has been formed, normalized data will be used to train the model with configured batch size and epoch obtained in GridSearchCV. After the training process completed, the model will be evaluated by measuring the RMSE and MAPE scores on test set. Model testing will be carried out 20 times and the best results will be taken from the model and stored in .h5 form. The best model also be converted into a .JSON file by using the TensorFlow JavaScript (TFJS) library so that the model can be used in Node.JS as the system backend.

3. Results and Discussion

The experiment was conducted on DELL Inspiron 7472 with 16GB RAM and intel Core-i5 8250u processor. Nvidia MX150 was utilized together with JupyterLab and Postman on Visual Studio code environment. The parameter results with the best level of accuracy obtained are using batch size of 30, epochs of 20, dropout rate of 0.3 or 30%, first LSTM layer of 32 units, and second LSTM layer of 32 units.

The structure of the prediction model uses two LSTM layers and a dropout of 30%. Hence, after going through the first layer of LSTM, then Dropout is run to eliminate 30% passed neurons randomly and then proceed to the next second LSTM layer. The purpose of using this dropout is to randomly remove some neurons used in training [13] in order to prevent overfitting. Overfitting happens when the network is able to predict the training data sample accurately but has poor performance and cannot generalize well to the validation and test data [17]. Details of the structure of the selected model used can be seen in figure 6.

```
selected_model = Sequential()
selected_model.add(LSTM(32, input_shape=(x_train.shape[1], x_train.shape[2]), return_sequences=True))
selected_model.add(Dropout(0.3))
selected_model.add(LSTM(32, return_sequences=False))
selected_model.add(Dense(y_train.shape[1]))
selected_model.compile(optimizer='adam', loss='mean_squared_error', metrics=['acc'])
selected_model.summary()
```

Figure 6. Selected model configuration

After completing the training, the next stage is to test the model and assess the model performance using the RMSE and MAPE criteria. It was found that the RMSE and MAPE scores for daily positive case are 27.149 and 53.111% respectively, while the RMSE and MAPE for daily death case are 22.678 and 0.165% respectively. The prediction graphs for both daily positive and death cases are shown in figure 7 and figure 8.

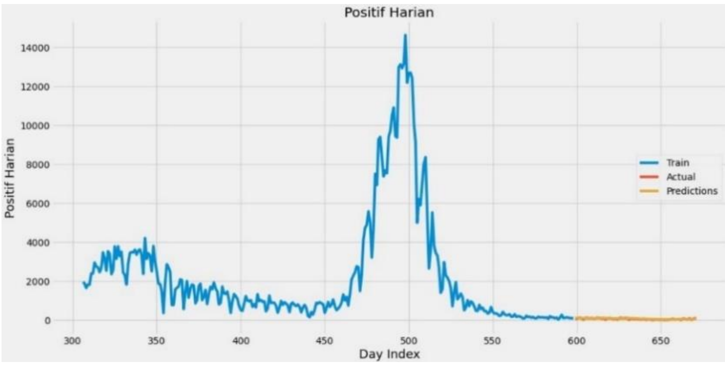

Figure 7. Prediction results for daily positive case

ISSN 2723-6471

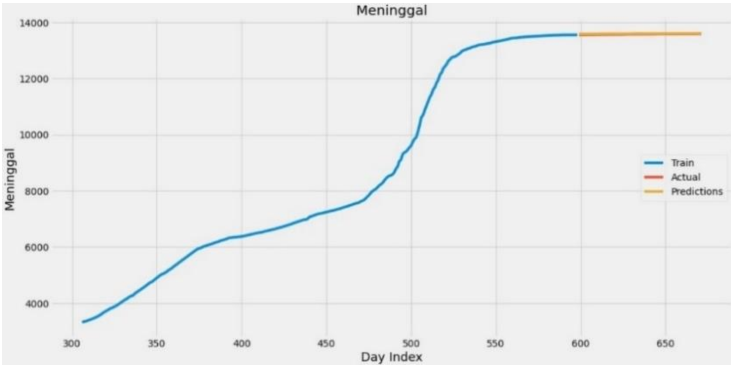

Figure 8. Prediction results for daily death case

After the backend has been created and prediction model developed, the next step is the creation of a frontend website that will display COVID-19 prediction data using React.JS. The data display will be divided into two parts, namely in graphic and in tabular form as well as special calculations for determining the PPKM level by using the latest seven daily positive and death data. We followed the PPKM level requirements as described in [18] and assumed the total population of Jakarta residents in 2021 reaching 10,609,681 people [19]. Figure 9 depicts the frontend system for PPKM level prediction based on the daily positive and death cases.

Figure 9. PPKM level prediction result

It is worthy to note that the discrepancy between daily positive and death cases prediction results might be caused by several factors, such as the usage of simple LSTM architecture with only two LSTM layers and one dropout layer, and the application of multivariate approach rather than univariate approach. Moreover, higher data distribution and variance in daily positive cases than the death cases also could affect the prediction results. Hence, although the proposed LSTM network with multivariate approach could be faster in terms of computing time, but the prediction results might be varied depend on the variance of historical data.

4. Conclusion

Based on the results of the research conducted, it can be concluded that the LSTM method can be used to predict the future daily positive and death cases of COVID-19 in Jakarta area. This in return could be used to determine the change of PPKM level for each region. However, the prediction results for daily positive case performed poorly compare to the daily death case. This might be rooted from the use of multivariate approach during the model development. Therefore, in future research, it is suggested to use univariate approach rather than multivariate to prevent interference between each data feature for this particular study case. Other Deep Learning methods, such as Gated Recurrent Unit (GRU), Bi-GRU, and w-GRU [13], [20] also can be implemented in future research.

5. Declaration

5.1. Author Contributions

Conceptualization: C.D. and S.H.; Methodology: S.H.; Software: C.D.; Validation: C.D. and S.H.; Formal Analysis: C.D. and S.H.; Investigation: C.D.; Resources: C.D. and S.H.; Data Curation: S.H.; Writing Original Draft Preparation: C.D.; Writing Review and Editing: S.H.; Visualization: C.D.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Liana and S. Hansun, “COVID-19 Vaccination Sentiment Analysis in Indonesia: A Data-Driven Approach,” *Int. J. Adv. Soft Comput. its Appl.*, vol. 16, no. 1, pp. 233–244, 2024.

[2] E. Haryono and A. Y. Harsari, “Taming Two Waves of the Covid-19 Pandemic: The Case of Indonesia,” *KnE Social Sciences*, vol. 7, no. 4, pp. 44–52, 2022, doi: 10.18502/kss.v7i4.10514.

[3] A. H. Shafaghi, F. R. Talabazar, A. Koşar, and M. Ghorbani, “On the Effect of the Respiratory Droplet Generation Condition on COVID-19 Transmission,” *Fluids*, vol. 5, no. 3, pp. 1-13, 2020, doi: 10.3390/fluids5030113.

[4] B. N. Harapan, T. Harapan, L. Theodora, and N. A. Anantama, “From Archipelago to Pandemic Battleground: Unveiling Indonesia's COVID-19 Crisis,” *J Epidemiol Glob Health*, vol.13, no. 4, pp. 591-603, Dec. 2023, doi: 10.1007/s44197-023-00148-7.

[5] S. Hansun, A. Suryadibrata, R. Nurhasanah, and J. Fitra, "Tweets Sentiment on PPKM Policy as a COVID-19 Response in Indonesia," *Indian J. Comput. Sci. Eng.*, vol. 13, no. 1, pp. 51–58, Feb. 2022, doi: 10.21817/indjcse/2022/v13i1/221301302.

[6] A. K. Ilafi, A. N. Fadhilah, and L. Jowanti, "Analysis of Government Policy in Handling Covid-19 in Indonesia," *Proc. Int. Conf. Data Sci. Off. Stat.*, vol. 2021, no. 1, pp. 498–506, Jan. 2022, doi: 10.34123/icdsos.v2021i1.152.

[7] A. Sulasikin, Y. Nugraha, J. Kanggrawan, and A. L. Suherman, "Forecasting for a Data-Driven Policy Using Time Series Methods in Handling COVID-19 Pandemic in Jakarta," *Proc. Int. Smart Cities Conf.*, vol. 2020, pp. 1-6, Oct. 2020, doi: 10.1109/ISC251055.2020.9239066.

[8] M. Elsaraiti and A. Merabet, "A Comparative Analysis of the ARIMA and LSTM Predictive Models and Their Effectiveness for Predicting Wind Speed," *Energies*, vol. 14, no. 20, pp. 1-16, 2021, doi: 10.3390/en14206782.

[9] B. Lindemann, T. Muller, H. Vietz, N. Jazdi, and M. Weyrich, "A Survey on Long Short-Term Memory Networks for Time Series Prediction," *Procedia CIRP*, vol. 99, pp. 650-655, 2021, doi: 10.1016/j.procir.2021.03.088.

[10] S. Hansun, F. P. Putri, A. Q. M. Khaliq, and H. Hugeng, "On searching the best mode for forex forecasting: bidirectional long short-term memory default mode is not enough," *IAES Int. J. Artif. Intell.*, vol. 11, no. 4, pp. 1596–1606, Dec. 2022, doi: 10.11591/ijai.v11.i4.pp1596-1606.

[11] A. W. Ramadhan, D. Adytia, D. Saepudin, S. Husrin, and A. Adiwijaya, "Forecasting of Sea Level Time Series using RNN and LSTM Case Study in Sunda Strait," *Lontar Comput. J. Sci. Inf. Technol.*, vol. 12, no. 3, pp. 130–140, Oct. 2021, doi: 10.24843/LKJITI.2021.v12.i03.p01.

[12] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," *Procedia Comput. Sci.*, vol. 170, no. 1, pp. 1168–1173, 2020, doi: 10.1016/j.procs.2020.03.049.

[13] S. Hansun and F. P. Putri, "Weighted Deep Learning Approach for Better Forecasting," *J. Appl. Sci. Eng.*, vol. 27, no. 4, pp. 2421–2428, 2023

[14] S. Hansun, V. Charles, T. Gherman, C. R. Indrati, and Subanar, "A Tuned Holt-Winters White-Box Model for COVID-19 Prediction," *Int. J. Manag. Decis. Mak.*, vol. 20, no. 3, 2021, doi: 10.1504/IJMDM.2021.10034422.

[15] N. F. Omran, S. F. Abd-el Ghany, H. Saleh, A. A. Ali, A. Gumaei, and M. Al-Rakhami, "Applying Deep Learning Methods on Time-Series Data for Forecasting COVID-19 in Egypt, Kuwait, and Saudi Arabia," *Complexity*, vol. 2021, no. 1, pp. 1–13, Mar. 2021, doi: 10.1155/2021/6686745.

[16] P. Sugiartawan, A. A. Jiwa Permana, and P. I. Prakoso, "Forecasting Kunjungan Wisatawan dengan Long Short Term Memory (LSTM)," *J. Appl. Inf. Syst. Comput.*, vol. 1, no. 1, pp. 43–52, Sep. 2018, doi: 10.33173/jsikti.5.

[17] X. Ying, "An Overview of Overfitting and its Solutions," *J. Phys. Conf. Ser.*, vol. 1168, no. 2, pp. 1-6, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.

[18] S. HR, R. Malli, S. Sumiati, H. Hasnawati, and A. Nursyam, "Implementation of PPKM in Social Economic Change Perspectives of Daily Labor and Workers in the New Normal Era," *Devot. J. Res. Community Serv.*, vol. 3, no. 3, pp. 164–174, Jan. 2022, doi: 10.36418/dev.v3i3.103.

[19] D. K. N. Rachim, A. Firdaus, and A. G. W. Saputro, "Analysis of the Impact of Population Growth in DKI Jakarta Using Logistic Model," *J. Edu. Math.*, vol. 5, no. 1, pp. 69–78, Jun. 2022, doi: 10.21043/jpmk.v5i1.14276.

[20] S. S. Jumaa, "Studying Electricity Load Forecasting and Optimizing User Benefits with Smart Metering," *J. Appl. Data Sci.*, vol. 4, no. 4, pp. 490–503, Dec. 2023, doi: 10.47738/jads.v4i4.147.