Indonesian Journal of Applied Mathematics and Statistics (IdJAMS)

Volume 1 Issue 1 June, 2024, pp. 51-64
https://idjams.com/index.php/jms/index

Predicting the Spread of COVID-19 in Riau Province Using the Recurrent Neural Network Method

Ivan Daniel Siringo Ringo[1], Muhammad Abi Hanif[2], Riska Agustina[3], Putri Sari[4], Adilla Anisa Fitri[5], Rifky Fauzi[6,*]

[1,2,3,4,5,6] *Department of Mathematics, Faculty of Science, Institut Teknologi Sumatera, Jalan Terusan Ryacudu, Desa Way Hui, Kecamatan Jatiagung, Lampung Selatan, 35365, Indonesia*

[*]*Corresponding author: rifky.fauzi@ma.itera.ac.id*

Copyright: ©2024 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution-ShareAlike 4.0 International License-(CC-BY-SA) (https://creativecommons.org/licenses/by-sa/4.0/)

Article Info

Article history:
Received: June 2024
Revised: August 2024
Accepted: August 2024
Available online: August 2024

Abstract

The COVID-19 pandemic has significantly impacted the world, including Riau Province, Indonesia. Predicting case developments accurately is crucial for controlling the virus's transmission. The purpose of this study is to forecast COVID-19 cases in the province of Riau by employing a Recurrent Neural Network (RNN). Between March 2020 and August 2021, data on daily cases, deaths, and recoveries was collected. After testing the RNN model with a range of look-back values, the ideal values were found to be 20 days for daily cases and 10 days for daily recoveries and deaths. The model's ability to precisely capture the trend of real data was demonstrated by how closely it matched. Strong predictive performance was indicated by the resulting RMSE values, which were 435.31 for daily cases, 13.61 for daily deaths, and 331.95 for daily recoveries.

Keywords: COVID-19; Forecast; Recurrent Neural Network

I. Introduction

The coronavirus disease (COVID-19) is respiratory disease caused by severe acute respiratory syndrome coronavirus-2 (SARS -CoV-2), which is a type of coronavirus. In December 2019, the first case of COVID-19 was reported in Wuhan, Hubei province in China, the world health organization (WHO) has declared a contagious coronavirus disease (COVID-19) as a global pandemic in month March 2020. In a period of eleven months, the disease has spread rapidly to all corners of the world, including Indonesia. Since Therefore, several precautionary measures have been taken such as lockdown, rapid tests, wearing masks, as well as self-quarantine, has been implemented by countries to stop the spread COVID-19 pandemic [1].

Indonesia is a country with numbers fourth largest population globally, so some experts predict that a pandemic COVID-19 has a negative impact significant and longer pandemic periods compared to more populous countries A little. On March 2, 2020, President Joko Widodo announced 2 cases of COVID-19 in Indonesia for the first time and spread The Corona virus in Indonesia has increased significantly since the government announced the first case [2]. On April 2, 2020, United States announced there were 1,790 positive cases COVID-19, 113 people died, and 112 cases healed. After 9 months since its announcement the first case in Indonesia on December 9 2020, the total number of positive cases of COVID-19 in Indonesia has reached 592,900 cases, making Indonesia the country with the highest COVID-19 cases in Asia Southeast [3].

The COVID-19 pandemic has provided significant impact around the world, including in Indonesia. Riau Province is one of them areas that have been quite badly affected by the pandemic This. Attempts to predict developments COVID-19 cases in this region are increasing important in order to plan and

implement the steps effective prevention [4]. One of the methods can be used to make predictions Recurrent Neural Network (RNN). RNN is part of the algorithm capable machine learning to analyze time-series data with consider the temporal sequence of the data. This makes RNN very suitable for carry out forecasting on COVID-19 data, in where time patterns play an important role. In this research, the RNN method is applied to predict the number of daily cases, the number of daily deaths, and the number of recoveries daily in Riau Province [5].

By utilizing time series data COVID-19 from March 2020 to August 2021, this research aims to produce a model accurate predictions. The model will used to project the pandemic situation in the future, so that the authorities can make better decisions in deal with the spread of the virus [6]. This approach is expected to make a contribution significant in handling the pandemic in the Province Riau, as well as providing useful analysis tools for future research.

This research will evaluate performance RNN models with various look-back values, viz 10, 15, and 20, to determine which configuration most effective. The prediction results will be analyzed uses Root Mean Square Error (RMSE) metric for ensure the level of accuracy and reliability. Through this approach, it is hoped that found the optimal prediction model for help in dealing with COVID-19 in Riau Province [7].

II. Methods

A. Recurrent Neural Networks

The system used to predict the spread of COVID-19 in Riau province is using Recurrent Neural Network (RNN). The COVID-19 prediction process begins with stage of determining the input data used for predictions. Dependent variable used are daily cases, daily deaths and daily recovered. Recurrent neural network or Recurrent Neural Network (RNN) is a type of neural network architecture mocks whose processing is called repeatedly to process input which is usually data sequential [8]. RNN is included in the deep category learning because data is processed through many layers (layers) [9]. The purpose of this test is to determine whether the system can measure the performance of the built model is, therefore, tested with existing parameters. These parameters including changing the learning rate value and changing the epoch value to get the weight the ideal.

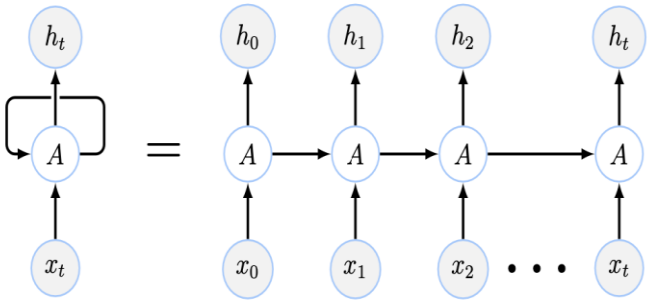


Figure 1. Recurrent Neural Network's Cell [9]

Figure 1 depicts the structure and unrolling of a Recurrent Neural Network (RNN) over time. On the left side of the image, we see a single RNN cell. The cell, denoted as *A*, takes two inputs: the current input x_t and the previous hidden state h_{t-1}. The output of the cell is the current hidden state ht, which is fed back into the cell in the next time step, creating a loop. This feedback mechanism allows the RNN to maintain a memory of previous inputs, which is crucial for processing sequential data.

The right side of the image shows the RNN "unrolled" over time. Unrolling means representing the network at multiple time steps, which helps in understanding how information is passed through the network. Each cell A at a time step t takes the current input x_t and the hidden state from the previous

time step h_{t-1}. The output of each cell is the hidden state ht, which is used as an input for the next time step. This unrolling continues for the length of the input sequence.

The process starts with an initial hidden state h_0, usually initialized to zero or learned as a parameter. At $t=0$, the first input x_0 is fed into the first cell along with h_0. The cell A processes these inputs and produces h_1. At $t=1$, the second input x_1 is fed into the second cell along with the hidden state from the previous time step h_1. This process is repeated for each subsequent time step: x_t and h_{t-1} are input to the cell A, which produces the new hidden state ht.

The hidden state ht at each time step encapsulates information from all previous inputs $x_0, x_1, \ldots, x_t$. This allows the RNN to maintain a memory of the sequence and use past information to inform the current output. RNNs process sequences of data one step at a time, maintaining a state that carries information across time steps. The weights and biases within the cell A are shared across all time steps, which helps in capturing patterns in the sequence.

Training RNNs involves a process called Backpropagation Through Time (BPTT), where the gradients are calculated through the unrolled network to update the parameters. This structure allows RNNs to be powerful for tasks involving sequential data, such as time series prediction, natural language processing, and speech recognition, where the order of the data points matters.

B. Data Preprocessing

The RNN algorithm process begins with enter data on the spread of COVID-19 into Python programming language [10]. Python is selected because of its flexibility and there are many libraries for the data mining process. This is possible developers to write more code easy to maintain, test, and develop. Then, data preprocessing is carried out with set dimensions and perform data scaling with the aim of preventing variable domination, optimize model performance, and improve interpretability [11]. As for formulas data scaling is

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

where X is the overall data, X_{min} is the smallest data, and X_{max} is the largest data. After data preprocessing is carried out, then form a new dataset from the results of preprocessing in the same way using look-back [12].

In system design in Recurrent Neural Network, there are several steps done. These steps include: designing architecture, setting initial weights initialize, initialize parameters, prepare input vector, output vector, and determine the target network [13]. Recurrent Neural Network is a type of neural network that has repetitive nature. This network is called recurrent because value of neurons in the previous hidden layer will be reused as input data. The neurons in the hidden layer will be stored in a special layer called the context layer. Values neurons in the context layer will be continuously updated until the RNN condition is met [13].

First, raw data is cleaned, normalized, and made ready for input into the model. This is commonly referred to as data preprocessing. The Recurrent Neural Network (RNN) architecture is then developed, specifying the layers, units, and parameters required to extract temporal dependencies from the data. The preprocessed data is used to train the RNN, which then uses back-propagation to update weights in order to minimize prediction error. The model is tested after training to assess its effectiveness and make sure it applies well to new data.

C. Model Evaluations

In this section, we focus on the evaluation of the performance of Recurrent Neural Networks (RNNs) using widely used error metric i.e. Root Mean Square Error (RMSE). This metric help quantify the accuracy of the predictions made by the RNN, offering insights into how well the model is performing..

The RMSE provides a measure of the differences between the predicted values and the actual values in our dataset. It is calculated by taking the square root of the average of the squared differences between the predictions and the actual observations. Mathematically, it is represented as

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

where $\hat{y}_i$ is predicted value, y_i is actual data, and n is number of observation. A lower RMSE indicates a closer fit of the model to the data, meaning that the model's predictions are close to the actual values.

D. Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is one of popular optimization techniques used in machine learning, particularly for training models like neural networks. Slightly different than gradient descent, SGD updates the model's parameters using only a single or a small batch (sampling) of data points at each step. This approach significantly reduces the computational cost and memory usage, making it well-suited for large datasets. Meaning SGD involves randomness into the optimization process, which can help the model escape local minima and potentially find a better global minimum.

The randomness in parameter updates also means that the convergence process in SGD is noisier compared to batch gradient descent. However, this noise can be advantageous as it provides opportunities for the model to explore a wider range of parameter space, potentially leading to better overall solutions. To balance the exploration and convergence, various strategies such as learning rate schedules and momentum are often employed. Learning rate schedules adjust the step size over time, typically decreasing it as the training progresses, to ensure the model stabilizes as it approaches the optimal solution. Momentum method helps to smooth out the updates by considering the past gradients, which accelerates convergence in the relevant direction and reduces oscillations.

III. Results and Discussion

Data used in this research is data on the distribution of COVID-19 from the Province Riau, which covers the time span from March 2020 to August 2021 as this represent the first wave in Indonesia. The data consists of three main features, namely the number of daily cases, number of daily deaths, and total daily healing (daily recovered). By Overall, 536 data entries were used in this study.

The Recurrent Neural Network (RNN) architecture that is offered is intended for time series forecasting, more especially for estimating how a phenomenon such as COVID-19 case counts will develop in the future. First, a 50-unit Simple RNN layer in the architecture processes sequential input data (e.g., previous case numbers across time) to capture temporal dependencies. The look_back parameter, which indicates how many previous time steps are used to forecast the future, defines the input to this layer. The outputs from the RNN layer are then further processed by a Dense layer with 20 units and ReLU activation, which adds non-linearity and improves the model's ability to identify complex interactions in the data.

Data on the spread of COVID-19 used is time-series data that cannot be direct processed into a Neural Network model. This matter occurs because the Neural Network is supervised learning, which requires input in the form of features and outputs. That's why time-series data will first formed into a dataset with how to create a look-back. In For this problem, look-backs 10, 15, and 20 were selected to analyze the magnitude of the error using metric RMSE [4].

Table 1. Data on the Distribution of COVID-19 in Indonesia

Date	Daily Case	Daily Death	Daily Recovered
02/03/2020	1	0	0
03/03/2020	0	0	0
04/03/2020	0	0	0
…	…	…	…

Date	Daily Case	Daily Death	Daily Recovered
…	…	…	…
…	…	…	…
16/08/2021	609	42	1684
17/08/2021	819	33	1540
18/08/2021	589	30	1098

In this study, we used three main features, namely the number of daily cases, number of daily deaths, and total daily healing (daily recovered). Data can be seen in Table 1. After that, a fitting process is carried out for each feature, namely processing the dataset that has been through preprocessing based on RNN architecture has been designed.

A. Model RNN Daily Case

In the fitting process using look-backs 10, 15, and 20, a Loss function graph is obtained for the three look-backs as shown in Figure 2-4.

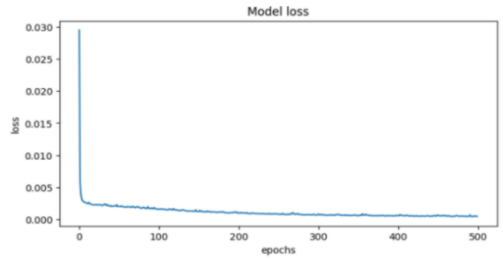


Figure 2. Training Loss of the RNN Model over 500 epochs for daily cases with look-back=10

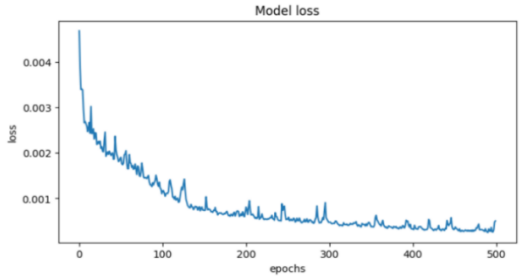


Figure 3. Training Loss of the RNN Model over 500 epochs for daily cases with look-back=15

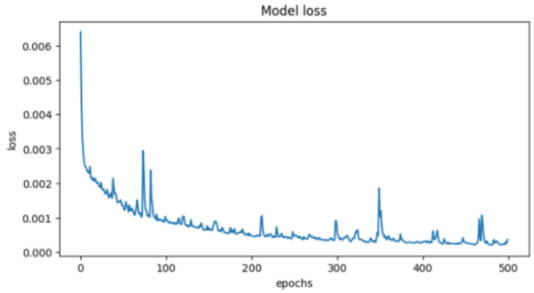


Figure 4. Training Loss of the RNN Model over 500 epochs for daily cases with look-back=20

The loss function from Figure 2-4 show comprehensive views of the RNN model's training progress over 500 epochs. Initially, there is a rapid decrease in the loss, especially in Figure 2 with look-back=20, indicating that the model is quickly learning and improving its performance during the early stages of training. This rapid reduction is typical as the model adjusts its weights from their initial random values. After approximately 100 epochs, the rate of decrease in the loss slows down and starts to stabilize, suggesting that the model is refining its performance and approaching an optimal state. This stabilization implies that the model is making smaller improvements with each epoch as it converges towards a minimum loss value.

Throughout the training process, a few noticeable spikes in the loss occur, particularly around epochs 300 and 400. These spikes could be attributed to various factors such as learning rate adjustments, model updates, or batch effects during training. Despite these fluctuations, the overall trend demonstrates a consistent decrease in loss, reaching a very low value by the end of the 500 epochs. This consistent downward trend indicates that the model has been effectively trained and that the optimization process has been successful.

The convergence of the loss towards a very low value suggests that the model is not overfitting and has generalized well to the training data. This is a positive sign, indicating that the model is likely to perform well on unseen data. To further enhance the model's training process, several improvements can be considered. The occasional spikes are seen as we use stochastic gradient descent for training the RNN model.

Moreover, from the figures it is also clear that the loss function with look-back=10 is flatter and the value converges as the epoch increases. This model uses fewer data input, so that fitting algorithm were facing smaller dimensional parameter estimation. Meaning that the model is much simpler that the others. The result can be seen in Figure 4-6 after inverse scaling the prediction results.

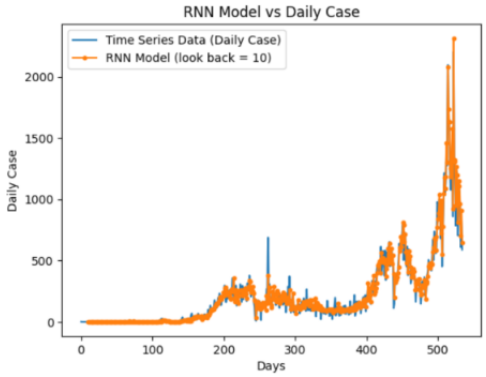


Figure 5. Comparison of Actual Daily cases (blue line) vs. Predicted Daily cases by RNN Model (orange line) with a Look-back Period of 10 Days

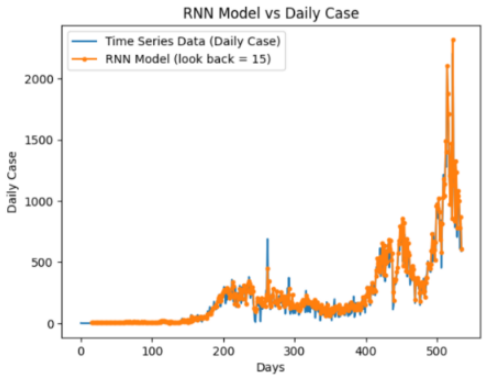


Figure 6. Comparison of Actual Daily cases (blue line) vs. Predicted Daily cases by RNN Model (orange line) with a Look-back Period of 15 Days

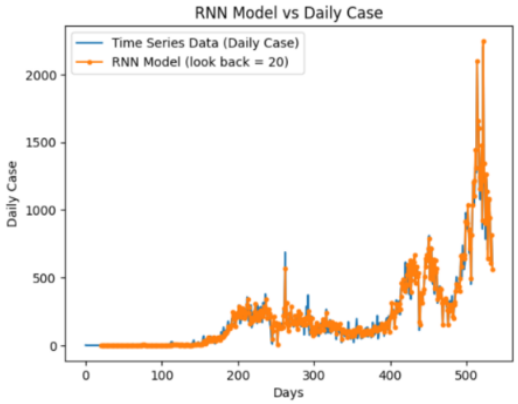


Figure 7. Comparison of Actual Daily cases (blue line) vs. Predicted Daily cases by RNN Model (orange line) with a Look-back Period of 20 Days

Based on these three figure, the model built is quite good in following the trend of the actual data. This is characterized by a model that follows data trends both when the trend is down and when the trend is up with quite appropriate time.

Next, calculations are carried out to get the RMSE, the RMSE for each look-back can be seen in the Table 2.

Table 2. RMSE Daily Case Calculation Results

Look-back	RMSE
10	437.154141
15	438.534049
20	435.309074

Because look-back=20 has the smallest RMSE, the RNN model with look-back = 20 was chosen as the best model. Next, a forecasting process was carried out for the next time step and a forecasting result was obtained with a value namely 959.07.

B. Model RNN Daily Death

In the fitting process using look-backs 10, 15, and 20, a Loss function graph is obtained for the three look-backs as shown in Figure 8-10.

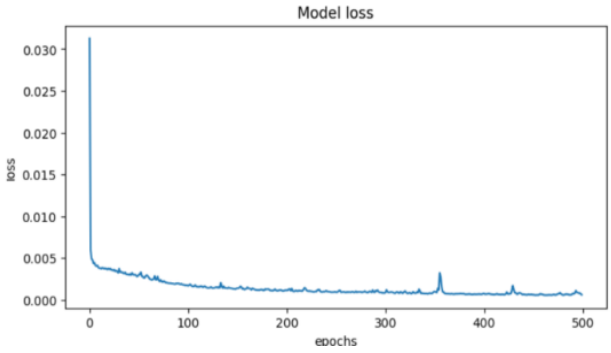


Figure 8. Training Loss of the RNN Model over 500 epochs for daily deaths with look-back=10

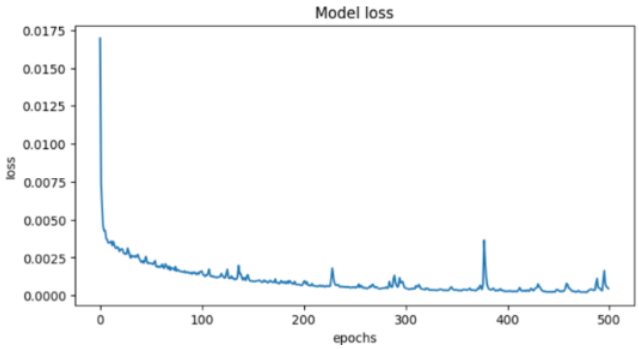


Figure 9. Training Loss of the RNN Model over 500 epochs for daily deaths with look-back=15

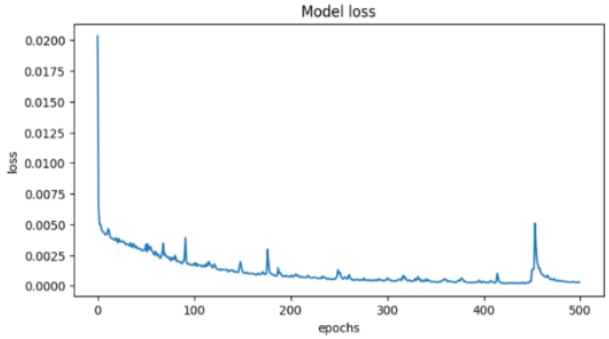


Figure 10. Training Loss of the RNN Model over 500 epochs for daily deaths with look-back=20

The three pictures above show that the loss function with look-back = 10 is flatter and the value converges as the epoch increases. After that, the scale is returned to see the prediction results. The graph of the prediction results after rescaling is as shown in the image below.

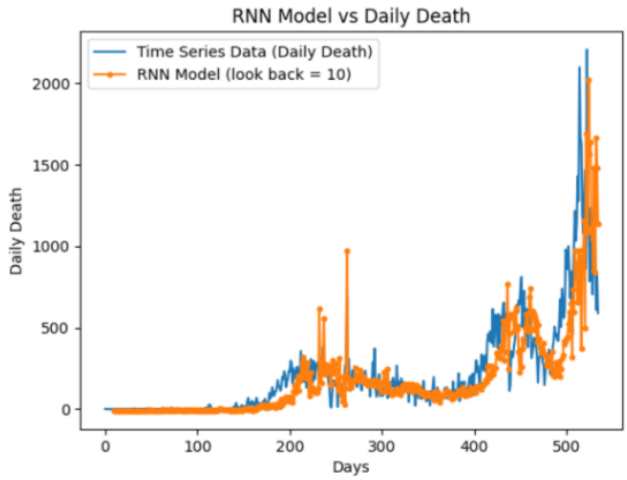


Figure 11. Comparison of Actual Daily cases (blue line) vs. Predicted Daily deaths by RNN Model (orange line) with a Look-back Period of 10 Days

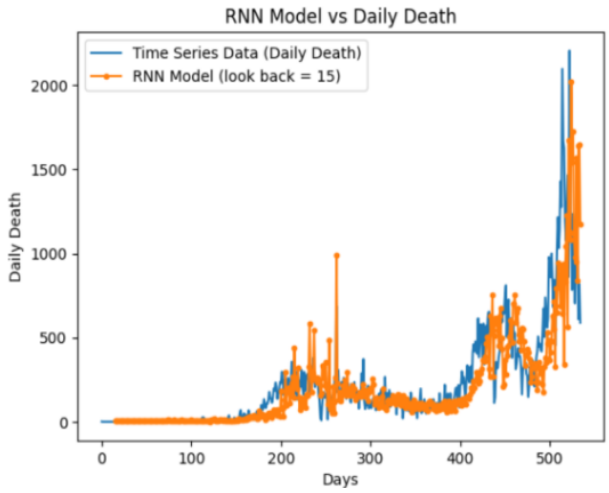


Figure 12. Comparison of Actual Daily cases (blue line) vs. Predicted Daily deaths by RNN Model (orange line) with a Look-back Period of 15 Days

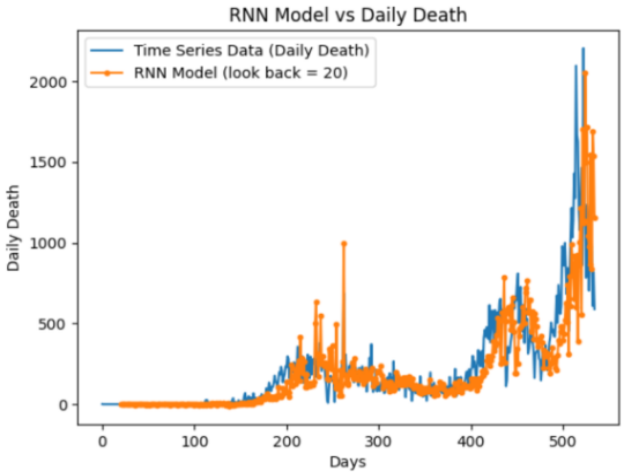


Figure 13. Comparison of Actual Daily cases (blue line) vs. Predicted Daily deaths by RNN Model (orange line) with a Look-back Period of 20 Days

Based on these three graphs, the model built is quite good. This is characterized by a model that follows data trends both when the trend is down and when the trend is up with quite appropriate momentum.

Next, calculations are carried out to get the RMSE, the RMSE for each look-back can be seen in the Table 3.

Table 3. RMSE Daily Death Calculation Results

Look-back	RMSE
10	13.609805
15	13.989217
20	13.803985

Because look-back = 10 has the smallest RMSE value, the RNN model with look-back = 10 was chosen. Unlike the preceding section's model for daily cases, the model for daily deaths performs better with a shorter look-back period. Specifically, a look-back period of 10 days yields the best results. Next, a

forecasting process was carried out for the next period and a forecasting result was obtained with a value of 41.5999.

C. Model RNN Daily Recovered

In the fitting process using look-backs 10, 15, and 20, loss functions are depicted for the three look-backs as shown in Figure 15-17.

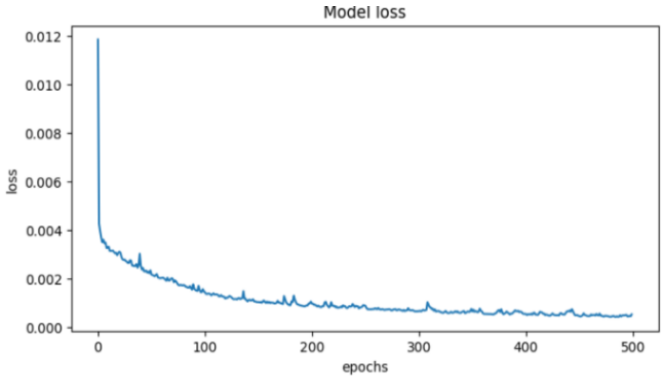


Figure 14. Training Loss of the RNN Model over 500 epochs for daily recovered with look-back=10

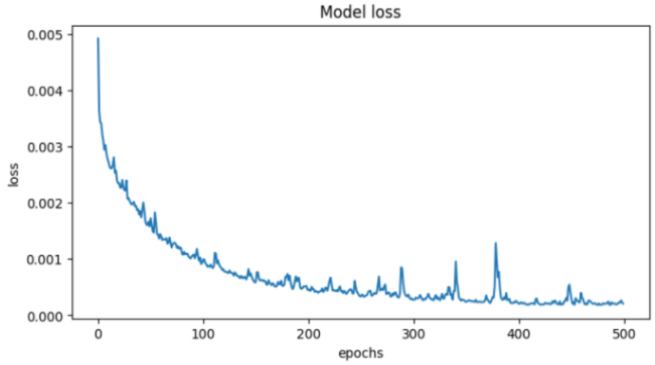


Figure 15. Training Loss of the RNN Model over 500 epochs for daily recovered with look-back=15

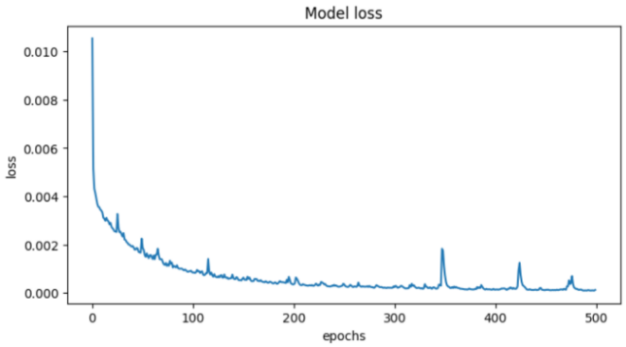


Figure 16. Training Loss of the RNN Model over 500 epochs for daily recovered with look-back=20

The three figures show that the loss function with look-back = 10 is flatter and the value converges as the epoch increases. After that, the scale is returned to see the prediction results. The predicted values for daily recovered after rescaling are as shown in Figure 18-20.

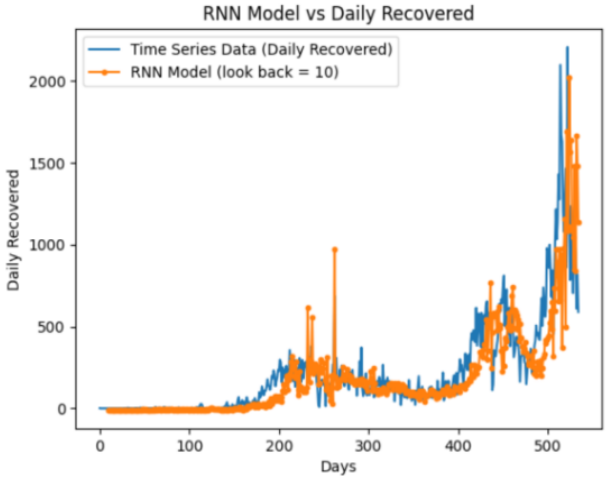


Figure 17. Comparison of Actual Daily cases (blue line) vs. Predicted Daily deaths by RNN Model (orange line) with a Look-back Period of 10 Days

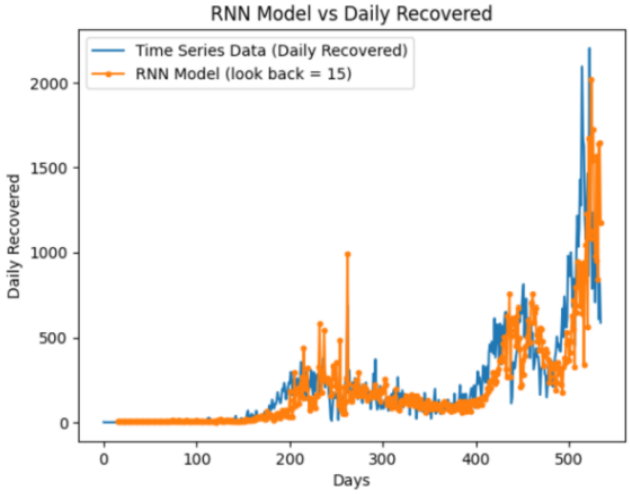


Figure 18. Comparison of Actual Daily cases (blue line) vs. Predicted Daily deaths by RNN Model (orange line) with a Look-back Period of 15 Days

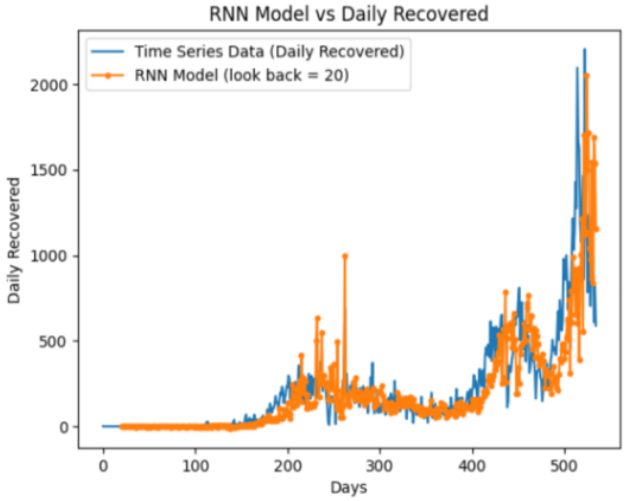


Figure 19. Comparison of Actual Daily cases (blue line) vs. Predicted Daily deaths by RNN Model (orange line) with a Look-back Period of 10 Days

Based on these three graphs, the model built is quite good. This is characterized by a model that follows data trends both when the trend is down and when the trend is up with quite appropriate momentum. Next, calculations are carried out to get the RMSE, the RMSE for each look-back can be seen in the Table 4.

Table 4. RMSE daily recovered calculation results

Look-back	RMSE
10	331.949041
15	345.364117
20	346.048650

Because look-back = 10 has the smallest RMSE, the RNN model with look-back = 10 was chosen. Next, a forecasting process was carried out for the next period and a forecasting result was obtained with a value of 676.34.

The study's practical limitation is from its dependence on data that could be inaccurate or incomplete as a result of underreported COVID-19 cases. Furthermore, the model's forecasts might not take into account abrupt changes in public health policies or virus mutations as the forecasts are based on historical data. Another difficulty with RNNs are its computational complexity, requiring for a lot of time and computational power that not all institutions might be able to provide.

Although RNNs are successfully used in this study to forecast the spread of COVID-19, there are still a few unresolved problems. For example, enhancing model accuracy when dealing with noisy or sparse data is a continuous problem. Moreover, combining adaptive learning techniques with real-time data sources may improve prediction reliability in quickly evolving circumstances. In order to better capture the complex nature of epidemic spread, hybrid models that combine RNNs with other machine learning techniques may be investigated in future studies.

IV. Conclusion

This study on the spread of COVID-19 in the Province of Riau, covering March 2020 to August 2021, utilized RNN models with look-back periods of 10, 15, and 20 days to predict daily cases, deaths, and recoveries. The model for daily cases performed best with a 20-day look-back, achieving the lowest RMSE value, while models for daily deaths and recoveries showed optimal performance with a 10-day look-back, also achieving the lowest error. Despite occasional spikes in the loss function due to stochastic gradient descent, all models effectively captured the data trends, demonstrating strong generalization and accurate prediction capabilities. These results highlight the importance of selecting appropriate look-back periods for different time-series features to enhance model's accuracy.

Acknowledgment

We would like to thank Department of Mathematics Institut Teknologi Sumatera for supporting the research, and the anonymous reviewers for providing constructive comments to improve the earlier version of the manuscript.

References

[1] A. A. Ghozi, A. Aprianti, A. D. P. Dimas, and R. Fauzi, "Prediction Analysis of Covid-19 Case Data in Lampung Province Using Recurrent Neural Network (RNN)," *Indonesian J. Appl. Math.*, vol. 2, no. 1, pp. 25–32, Apr. 2022, doi: https://doi.org/10.35472/indojam.v2i1.763.

[2] M. R. Davahli, W. Karwowski, and K. Fiok, "Optimizing COVID-19 vaccine distribution across the United States using deterministic and stochastic recurrent neural networks," *PLOS ONE*, vol. 16, no. 7, p. e0253925, Jul. 2021, doi: https://doi.org/10.1371/journal.pone.0253925.

[3] M. Habib and A. Dzakwan, "Memetakan Kesiapan Pemerintah Daerah dalam Menangani COVID-19," 2020. Accessed: Jun. 21, 2024. [Online]. Available: https://s3-csis-web.s3.ap-southeast-1.amazonaws.com/doc/CSIS_Commentaries_DMRU-027_Dzakwan.pdf

[4] A. Farhah, A. L. Prasasti, and M. W. Paryasto, "Implementasi Recurrent Neural Network dalam Memprediksi Kepadatan Restoran Berbasis LSTM," *J. Media Infor.*, vol. 5, no. 2, pp. 524–531, Apr. 2021, doi: https://doi.org/10.30865/mib.v5i2.2916.

[5] A. Rizal and S. Hartati, "Prediksi Kunjungan Wisatawan Dengan Recurrent Neural Network Extended Kalman Filter," *J. Ilmiah Ilmu Komputer Univ Udayana*, vol. X, no. 1, 2017, Accessed: Jun. 21, 2024. [Online]. Available: https://ojs.unud.ac.id/index.php/jik/article/download/39771/24168

[6] A. Hassan, I. Shahin, and M. B. Alsabek, "COVID-19 Detection System using Recurrent Neural Networks," *2020 Int. Conf. Commun. Comput. Cybersecur. Inform.*, Nov. 2020, doi: https://doi.org/10.1109/ccci49893.2020.9256562.

[7] G. S. Sianturi and R. Fauzi, "Pemodelan Kedalaman Laut Pada Perairan Selat Sunda dan Sekitarnya Menggunakan Neural Network" *Proc. Seminar Nasional Sains dan Teknologi "SainTek,"* vol. 1, no. 1, pp. 1–11, 2023, Accessed: Jun. 21, 2024. [Online]. Available: https://conference.ut.ac.id/index.php/saintek/article/view/2285

[8] R. L. Kumar, F. Khan, S. Din, S. S. Band, A. Mosavi, and E. Ibeke, "Recurrent Neural Network and Reinforcement Learning Model for COVID-19 Prediction," *Frontiers in Public Health*, vol. 9, Oct. 2021, doi: https://doi.org/10.3389/fpubh.2021.744100.

[9] L. A. Jati, "Prediksi Indeks Standar Pencemar Udara di Kota Bandar Lampung dengan Faktor Meteorologi Menggunakan Model Long Short-Term Memory(LSTM)," Tugas Akhir, Institut Teknologi Sumatera, 2023.

[10] A. P. Gema and D. Suhartono, "Recurrent Neural Network (RNN) dan Gated Recurrent Unit (GRU)," *School of Comput. Sci.*, Feb. 13, 2017. http://socs.binus.ac.id/2017/02/13/rnn-dan-gru/ (accessed Jun. 21, 2024).

[11] R. Qolbi, "Perencanaan Komunikasi Badan Penanggulangan Bencana Daerah Provinsi Riau dalam Menangani Penyebaran Covid-19 di Pekanbaru," *J. Ilmu Komunikasi (JKMS)*, vol. 9, no. 2, pp. 551–566, Jun. 2021, doi: https://doi.org/10.35967/jkms.v9i2.7445.

[12] K. C. Santosh, "COVID-19 Prediction Models and Unexploited Data," *J.. Med. Syst.*, vol. 44, no. 9, Aug. 2020, doi: https://doi.org/10.1007/s10916-020-01645-z.

[13] L. Louisa, R. Fauzi, and E. S. Nugraha, "Forecasting of Retirement Insurance Filled via Internet by ARIMA Models," *J. Actuar. Financ. Risk Manag.*, vol. 1, no. 1, pp. 1–8, Jun. 2022, doi: https://doi.org/10.33021/jafrm.v1i1.3672.

[14] D. P. SIAGIAN, "Perbandingan Radial Basis Function dan Recurrent Neural Network pada Prediksi Curah Hujan di Palembang," Skripsi, Universitas Sriwijaya, 2019. Available:https://repository.unsri.ac.id/23251/3/RAMA_55201_09021281320004_0006067406_0010058507_01_front_ref.pdf

[15] Muhamad Rizal Firmansyah, R. Ilyas, and Fatan Kasyidi, "Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network," *Proc. Industrial Research Workshop and National Seminar*, vol. 11, no. 1, pp. 488–495, Sep. 2020, doi: https://doi.org/10.35313/irwns.v11i1.2055.

[16] A. M. Siregar, J. H. Jaman, and A. Mufti, "Prediksi Kesehatan Masyarakat Indonesia Menggunakan Recurent Neural Network," *INTERNAL (Information System Journal)*, vol. 4, no. 1, pp. 28–34, Jun. 2021, doi: https://doi.org/10.32627/internal.v4i1.285.

[17] G. R. Kurnila, A. L. Nugraha, and A. Sukmono, "Analisis Spasial Tingkat Risiko Akibat Pandemi Coronavirus Disease 2019 (Covid – 19) (Studi Kasus : Kabupaten Indragiri Hulu, Provinsi Riau)", *Jurnal Geodesi Undip*, vol. 11, no. 1, pp. 1–10, Jan. 2022, doi: https://doi.org/10.14710/jgundip.2022.31815.

[18] Andrian *et al.*, "Prediksi Gelombang Corona Dengan Metode Neural Network," *J. Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 3, no. 2, pp. 102–107, Sep. 2020, doi: https://doi.org/10.9767/jikomsi.v3i2.74.