Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol 6, No.1, March 2023, pp. 98 – 106
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Small Timescaled Data for Covid-19 Prediction with RNN-LSTM in Tangerang Regency

[1]Sagita Sasmita Wijaya, [2]Marlinda Vasty Overbeek
[1,2]Department of Informatics, Universitas Multimedia Nusantara
Email: [1]sagitasasmita@gmail.com, [2]marlinda.vasty@umn.ac.id

Article Info
Article history:
Received Jan 31th, 2023
Revised Mar 13th, 2023
Accepted Mei 26th, 2023

Keyword:
Covid-19
LSTM
RNN
Small Time scaled Data
Tangerang Regency

ABSTRACT
Throughout the pandemic, many people have become familiarised with the new type of virus that has been spreading throughout the world, called the Coronavirus. On the 2nd of March, the year 2020, the Indonesian government had announced the identification of first Covid-19 case in Indonesia. With the arrival of Covid-19, and its spreading across all the provinces of Indonesia, the number of positive cases keeps growing even in the present day. Tangerang Regency is one of the areas that has opaqued citizens in the Banten Province. The purpose of this research is to discuss how to predict the sum of Covid-19 cases in the Tangerang Regency using the RNN-LSTM method. Although this method is very eloquent if used to perform a sequential task, its complexity and loss of gradient can make this model difficult to be trained, hence resulting in the use of the Long Short-Term Memory (LSTM) to reduce these weaknesses and help the RNN to look back on past data. This research uses Python as the programming language and Jupyter Notebook for the visualization of the results of the prediction. Therefore, the prediction model has been evaluated using various computational methods, such as RMSE with its error percentage of 0.05, and MSE and MAE with the same error percentage of 0.03 with the loss of their models being 9.6793e-04.

Corresponding Author:
Marlinda Vasty Overbeek
Department of Informatics,Faculty of Engineering and Informatics
Universitas Multimedia Nusantara,
Jl. Scientia Boulevard, Curug Sangereng, Kec. Klp. Dua, Kabupaten Tangerang, Banten 15810
Email: marlinda.vasty@umn.ac.id
DOI: http://dx.doi.org/10.24014/ijaidm.v6i1.21676

1. INTRODUCTION

Coronavirus or Covid-19 is one out of a large family of various viruses such as MERS and SARS. These viruses have an effect similar to the common flu, on civilization, until the people affected by the virus start experiencing respiratory problems that are more severe. In the year 2019, a variant of the Coronavirus had been found spreading widely in Wu Han, China, known as ‘severe acute respiratory syndrome coronavirus 2’ with the name ‘Coronavirus Disease 2019’ (Covid-19). In the middle of the year 2020, a worldwide organization by the name of ‘World Health Organization’ (WHO) affirmed that all the countries around the world must declare a state of pandemic simultaneously to break the spread of the virus, which had already begun spreading [1].

On the 2nd of March 2020, the Indonesian government had announced the appearance of the first case of Covid-19 in Indonesia, specifically in Tangerang Regency area, which is one of the areas that has opaqued citizens that support the nation’s capital as well. Hence, looking into the individual citizens who arrive and depart from Tangerang Regency is one of the most important concerns, considering that the number of Covid-19 cases and the number of suspected cases is increasing. ICU capacity and isolating rooms in hospitals reaching their limits is a great matter of concern for the whole city [2]. Therefore, to tackle these concerns, using deep learning is one of the best solutions. When inspecting the data manually it will be observed that this research uses the official dataset from Tangerang Regency’s official website, which will be converted into

Microsoft Excel. Manually crawling data is a method that is used to handle, discover, and explore data that is available in the World Wide Web [3]. Data crawling can be done by using machine learning, which is also a computational method that uses previous experience to grow and develop [4].

One method of machine learning is the Neural Network. This method can emulate a neural network that exists in a human, and it also has an input layer and an output layer comprising of a few units of neurons and it has an activation function to determine the result of these units. This method can moreover be improved by using preparatory information; thus, the more information utilized, the more superior the execution of the Neural Network. In any case, the Neural Network is restricted within a number of layers, although having more layers is advantageous, as the more layers there are, the higher capacity of the Neural Network [19]. Hence, to overcome this, Deep Learning was developed [5]. Deep Learning is a computational method that usually has many phases of techniques to represent data with various kinds of abstract reports. It also able to find complex structures in large-scale data structures using backpropagation. Thus, it can create a sophisticated result to solve problems that the AI community would not have been able to solve for several years, in advance [6].

One technique of deep learning that include a variety of Artificial Neural Networks is called Recurrent Neural Network (RNN), which is a technique that can basically be described as, creating a network topology to demonstrate a sequential and time-series related set of data [7]. In addition, this method has two inputs, where the first input is the current input while the second input is the input from the recent past. It also has an important feature called 'hidden state', which is able to memorize some information about the order of the existing data while its memory restores all the information about what has been calculated [8]. The use of Recurrent Neural Network has been practical for several previous research work, such as: prediction of multi-time steps by inputting data to be used as part of the learning process using RNN [9], evaluation of word translation issues [10], abbreviation of words in an information report [11], classification of quail egg quality [12], prediction of cement sales [13], and lastly an implementation of chatbots on a new student enrolment system [14].

There has been similar research done, specifically, the prediction of the Covid-19 patient count with the Karawang Regency case study. This research used the K-Nearest Neighbours method to predict Covid-19 cases with a superb and satisfactory performance result, however, it had insufficient predictive results for a few days. These results tended to stagnate, as the dataset used was still quite small. Nevertheless, its best performance was seen by adding 15 days' worth of data to the dataset [15]. Therefore, RNN can be used to develop a form of memory that is capable to help assume patterns with its dependence on previous patterns. It can also be used with a complex layer pattern, to expand the range of its very solid pixel [16]. Therefore, RNN is a preferable example of Long Short-Term Memory (LSTM), as its memory is able to recall all the information when building upon pre-existing proficiency [17]. In addition, RNN, has the capability to remember all the knowledge that has been previously inputted. This can be used advantageously, as it can predict the number of Covid-19 sufferers for data in a small-scale time series, in the hopes of helping the government and society fight Covid-19.

Based on previously discussed context, the overview of the problem at hand can be worded as follows; how to implement the Recurrent Neural Network (RNN) technique to predict the spread of Covid-19 using data within a small timescale, and how to measure its performance and predict the number of Covid-19 patients using a Recurrent Neural Network (RNN) algorithm for data within a small timescale. There are limitations that come into consideration when regarding the previously discussed problem, such as the fact that the dataset being used in this research (available on the official website of Tangerang Regency, https://covid19.tangerangkab.go.id/sebaran-data) has been calculated in the period from the 24th of June, 2021 to the 28th of February, 2022. The data used is regarding the confirmed number of Covid-19 patients and has been organised by date.

Therefore the objectives of this research are: to implement the Recurrent Neural Network (RNN) to predict the sum of people affected by the Covid-19 virus from the point in time when it began to spread until the present moment, to measure the performance of this prediction using the Recurrent Neural Network (RNN) algorithm, and to measure the accuracy of the results from the algorithm while focusing on a small dataset. In addition, based on the research objectives preciously described, the benefit of this research is that it can help predict how Covid-19 spreads within the Tangerang Regency, so that the society and the government can carry out disaster mitigation.

2. RESEARCH METHOD

In this research, there are various steps that will be accomplished such as, crawling the dataset, pre-processing, building the model and lastly, evaluating the model. The relevant flowchart system can be seen in Figure 1.

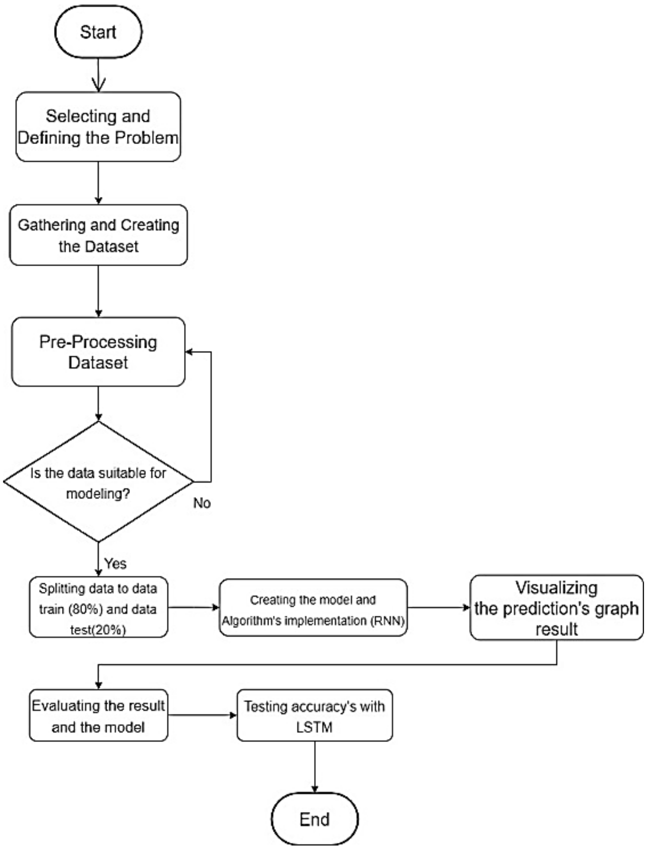

Figure 1. Flowchart System

2.1. Selecting and Identifying the Issue

The research will begin by identifying and focusing on the problem, which is to predict Covid-19 cases in the Tangerang Regency, and this will be resolved by using RNN-LSTM. Therefore, a model will be built to predict the results and then evaluate them, using an accuracy examiner and an error rate calculation. After identifying the problem, the research will continue by gathering and creating the dataset that needs to be collected.

2.2. Gathering and Creating the Dataset

In the following step, the dataset used in this research is taken from the official website of the Tangerang Regency (https://covid19.tangerangkab.go.id/sebaran-data). Data is collected every single day and measure variable called the 'sum of confirmed cases' is created. After collecting the data, it will be converted into the form of an .xlsx file and must be gathered first in order to be able to process it using the Python programming language.

2.3. Pre-Processing

Within the pre-processing step, all the necessary libraries will be imported, and the data will be normalized using the Min Max scaler. Before using this data, it must be ensured that there are no NaN (Not a Number) or blank data within the rows and columns of the data. To accommodate its used variables, there will be new variables created in the form of a matrix after all the pre-processing has been completed. The modelling of the algorithm will be continued.

2.4. Creating the Model and Algorithm's Implementation

Once the pre-processing has been finished, the next step is to create a model using RNN-LSTM, and the pre-processed data will be divided, with 80% of the data being used in training and 20% of the data being used for testing. The reason that the training data is of a larger percentage is because the RNN needs a large quantity of data for training, to increase its accuracy. Hence, the model will be using an LSTM layer to support the RNN, when predict Covid-19 patients. After splitting the dataset into 'data train' and 'data test', these data inputs will be reshaped into 3D inputs called 'samples', 'time step' and 'features'. In addition, before making the model there will be a variable called ‘look back’, that will be able to look into previous data and variable models to accommodate some of the layers, such as LSTM layer, dropout layer, and dense layer.

Therefore, after deciding on the hidden layer and input shape on the LSTM layer, the dropout later will be increased to 20% to prevent overfitting, while adding a dense layer to clarify the output that will be displayed, and an activation function to support the neural network in understanding the complex pattern that exists within the data. The optimizer used in this research is the Adam optimizer, which is an additional optimization algorithm that replaces the classical stochastic gradient descent, to affect the weight of its network based on training data.

2.5. Evaluating Model

Once the model has been successfully created, the next step is to evaluate the model and its error percentage using the Mean Squared Error (MSE) and LSTM. To evaluate the error rate, MSE will be used, while LSTM will be used to evaluate the accuracy. However, it needs tuning to minimalize its complexity by deciding the epochs, batch size and verbose. Epochs will be used to indicate the sum of paths from all the data in the training dataset, while batch size is a hyperparameter to define the sum of the sample that is needed to be finished before it can affect the parameters that are inside the internal model. Verbose will be used to indicate the result of the sum of the epochs. In addition, LSTM can be interpreted as a type of RNN that is able to learn to predict problems in a sequential and structured manner. Hence, an algorithm that has an area complexity in Deep Learning is also called LSTM [18]. The diagram of the LSTM cell can be seen below in Figure 2.

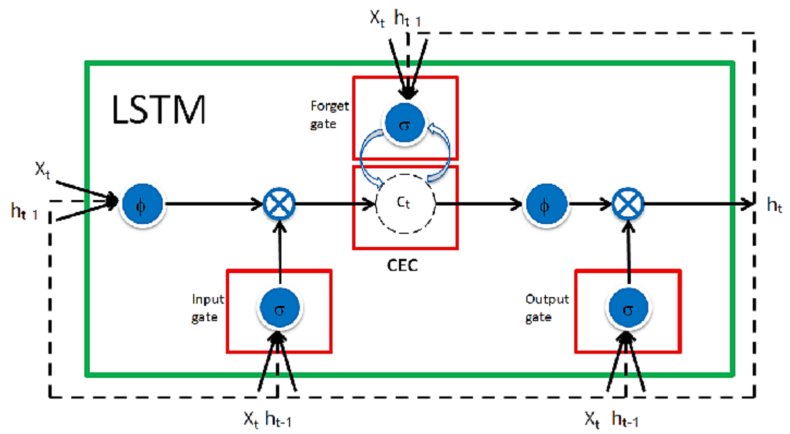

Figure 2. LSTM diagram cell

The figure above shows φ, σ and ⊗, as hyperbolic-tangent functions, while the sigmoid function can distinguish a singular value operation as max or min and a multiplication function. The LSTM formula can be written down mathematically below.

$$\underline{c}_t = \sigma(W_f \underline{I}_t)\underline{c}_{t-1} + \sigma(W_i \underline{I}_t)\phi(W_{in}\underline{I}_t), \tag{1}$$

$$\underline{h}_t = \sigma(W_o \underline{I}_t)\phi(\underline{c}_t), \tag{2}$$

The formula above explains that $\underline{\mathbf{c}}_{\mathbf{t}} \in \mathbb{R}^{\mathbf{N}}$ in the vector column of $\underline{\mathbf{I}}_{\mathbf{t}} \in \mathbb{R}^{(\mathbf{M}+\mathbf{N})}$, is a series of new inputs of $\underline{\mathbf{X}}_{\mathbf{t}} \in \mathbb{R}^{\mathbf{M}}$, and that the previous output $\underline{\mathbf{h}}_{\mathbf{t-1}} \in \mathbb{R}^{\mathbf{N}}$, can also be written in $\underline{\mathbf{I}}_{\mathbf{t}}^{\mathbf{T}} = [\underline{\mathbf{X}}_{\mathbf{t}}^{\mathbf{T}}, \underline{\mathbf{h}}_{\mathbf{t-1}}^{\mathbf{T}}]$. Hence, $\mathbf{W_f}$, $\mathbf{W_i}$, $\mathbf{W_o}$, and $\mathbf{W_{in}}$ is the weight of the matrix for the forget target, input gate, and output gate. It can then be assumed that $\underline{\mathbf{c}}_{\mathbf{0}} = \underline{\mathbf{0}}$ is the hidden state vector of the LSTM, which can be reduced to the calculation shown below.

$$\underline{\mathbf{c}}_{\mathbf{t}} = \underbrace{\sum_{k=1}^{t} \left[\prod_{j=k+1}^{t} \sigma(W_f \underline{I}_j) \right]}_{\textit{forget gate}} \sigma(W_i \underline{I}_k)\phi(W_{in}\underline{I}_k) \tag{3}$$

By using $\mathbf{f}(\cdot)$, present in formula number 3 shown above, it can be defined as a hyperbolic tangent, and the output results from LSTM can be seen below:

$$\underline{h}_t^{LSTM} = \sigma(W_o\underline{I}_t)\phi\left(\sum_{k=1}^{t}\underbrace{\left[\prod_{j=k+1}^{t}\sigma(W_f\underline{I}_j)\right]}_{forget\ gate}\sigma(W_i\underline{I}_k)\emptyset(W_{in}\underline{I}_k)\right). \quad (4)$$

From the formula above, it could be stated that W_c^{t-k} and $\prod_{j=k+1}^{t}\sigma(W_f\underline{I}_j)$ perform as the same memory for the LSTM. Therefore, there is a special case study whereas the memory length of the LSTM can transcend regardless of the parameter model i.e., W_c , W_{in},which can be seen in formula number 5 shown below.

$$\exists W_f \text{ s.t. } \min\left|\sigma(W_f\underline{I}_j)\right| \geq \sigma_{max}(W_c), \forall\sigma_{max}(W_c) \in [0,1), \quad (5)$$

Which turned into

$$\left|\prod_{j=k+1}^{t}\sigma(W_f\underline{I}_j)\right| \geq \sigma_{max}(W_c)^{t-k}, t \geq k\,. \quad (6)$$

Based on formula number 5, the impact input of $\underline{I}_k$, it can be said that there will always be an LSTM with a longer memory capacity, as the output of the LSTM can last longer.

3. RESULTS AND ANALYSIS

Within the implementation of the model, there are various test scenarios which will be demonstrated, and its results evaluated. The test scenarios that will be accomplished in this research are as follows:

1. The ‘used data’ test has a ratio of 80:20 with 80% of total case data used in training and 20% of total case data used in testing, while it contains the sum of confirmed Covid-19 cases in Tangerang Regency from June 2021 to February 2022.
2. Implement epoch tuning within 100 epochs, due to the error percentage and its resulting diagram being higher.

In addition, there will be an examination done by changing various parameter values such as varying the ‘look back’ parameter, increasing or decreasing the LSTM unit, switching the activation function and lastly, modifying the parameter of its learning rate for the Adam optimizer within the testing of the model. The evaluation result will be taken from the best performance out of the three examinations, shown in Table 1.

Table 1. The examination result with ratio of 80:20 using ReLU

Look Back	LSTM Units	Epochs	Learning Rate	Loss	MSE	RMSE	MAE
15	15	10	0.01	0.0034	0.01(train) 0.00(test)	0.10(train) 0.01 (test)	0.01(train) 0.00(test)
			0.001	0.0104	0.05(train) 0.00(test)	0.23(train) 0.01 (test)	0.05(train) 0.00 (test)
		100	0.01	0.0018	0.00(train) 0.00(test)	0.06(train) 0.01(test)	0.01(train) 0.00(test)
			0.001	0.0034	0.01(train) 0.00 (test)	0.10(train) 0.01 (test)	0.01(train) 0.00 (test)
	50	10	0.01	0.0036	0.01 (train) 0.00 (test)	0.08(train) 0.01 (test)	0.01(train) 0.00 (test)
			0.001	0.0075	0.04 (train) 0.00 (test)	0.20(train) 0.00 (test)	0.04(train) 0.00 (test)
		100	0.01	0.0013	0.00 (train) 0.00 (test)	0.04(train) 0.01(test)	0.00(train) 0.00 (test)
			0.001	0.0031	0.01 (train) 0.00 (test)	0.09(train) 0.01 (test)	0.01(train) 0.00 (test)
30	15	10	0.01	7.8865e-04	0.02 (train) 0.00 (test)	0.14(train) 0.00 (test)	0.02(train) 0.00 (test)
			0.001	0.0012	0.05 (train) 0.00 (test)	0.22(train) 0.00 (test)	0.05(train) 0.00 (test)
		100	0.01	1.9747e-04	0.02 (train) 0.00 (test)	0.15(train) 0.00 (test)	0.02(train) 0.00 (test)
			0.001	2.2683e-04	0.02 (train) 0.00 (test)	0.15(train) 0.00 (test)	0.02(train) 0.00 (test)
	50	10	0.01	5.5181e-04	0.03 (train)	0.17(train)	0.03(train)

Look Back	LSTM Units	Epochs	Learning Rate	Loss	MSE	RMSE	MAE
					0.00 (test)	0.00 (test)	0.00 (test)
			0.001	8.1413e-04	0.03 (train) 0.00 (test)	0.18(train) 0.00 (test)	0.03(train) 0.00 (test)
			0.01	4.7815e-04	0.02 (train) 0.00 (test)	0.14(train) 0.01 (test)	0.02(train) 0.00 (test)
		100	0.001	**1.1943e-04**	0.02 (train) 0.00 (test)	0.15(train) 0.00 (test)	0.02(train) 0.00 (test)

The Table 1 shown above can be used to explain the 'look back' parameters being used, which are 15 and 30, with LSTM units being 15 and 50, while the epochs are 10 and 100 with a learning rate of 0.01 and 0.001. The error percentages obtained are an MSE of 0.05, RMSE of 0.23 and MAE of 0.5, with the lowest loss being 1.1943e-04. Once the examination model using the ReLU activation function has been achieved, the following step is to the perform the same examination with different activation functions. Hence, Table 2 shown below signifies the examination result obtained when using the Tanh activation function.

Table 2. The examination result with ratio of 80:20 using Tanh

Look Back	LSTM Units	Epochs	Learning rate	Loss	MSE	RMSE	MAE
15	15	10	0.01	0.0066	0.02 (train) 0.00 (test)	0.13 (train) 0.00 (test)	0.02 (train) 0.00 (test)
			0.001	0.0159	0.05 (train) 0.00 (test)	0.23 (train) 0.00 (test)	0.05 (train) 0.00 (test)
		100	0.01	0.0021	0.00 (train) 0.00 (test)	0.07 (train) 0.01(test)	0.00 (train) 0.00 (test)
			0.001	0.0035	0.01 (train) 0.00 (test)	0.11 (train) 0.00 (test)	0.01 (train) 0.00 (test)
	50	10	0.01	0.0036	0.01 (train) 0.00 (test)	0.12 (train) 0.00 (test)	0.01 (train) 0.00 (test)
			0.001	0.0066	0.05 (train) 0.00 (test)	0.21 (train) 0.00 (test)	0.05 (train) 0.00 (test)
		100	0.01	0.0026	0.00 (train) 0.00 (test)	0.06 (train) 0.01(test)	0.00 (train) 0.00 (test)
			0.001	0.0041	0.01 (train) 0.00 (test)	0.11 (train) 0.00 (test)	0.01 (train) 0.00 (test)
30	15	10	0.01	3.9229e-04	0.03 (train) 0.00 (test)	0.18 (train) 0.00 (test)	0.03 (train) 0.00 (test)
			0.001	0.0010	**0.06** (train) 0.00 (test)	**0.24** (train) 0.00 (test)	**0.06** (train) 0.00 (test)
		100	0.01	1.7815e-04	0.02 (train) 0.00 (test)	0.14 (train) 0.01 (test)	0.02 (train) 0.00 (test)
			0.001	1.4521e-04	0.02 (train) 0.00 (test)	0.16 (train) 0.00 (test)	0.02 (train) 0.00 (test)
	50	10	0.01	2.6457e-04	0.03 (train) 0.00 (test)	0.17 (train) 0.00 (test)	0.03 (train) 0.00 (test)
			0.001	9.8359e-04	0.04 (train) 0.00 (test)	0.20 (train) 0.00 (test)	0.04 (train) 0.00 (test)
		100	0.01	8.0615e-05	0.02 (train) 0.00 (test)	0.14 (train) 0.01 (test)	0.02 (train) 0.00 (test)
			0.001	**6.0553e-05**	0.02 (train) 0.00 (test)	0.15 (train) 0.00 (test)	0.02 (train) 0.00 (test)

Therefore, it can be seen that this second examination using the Tanh activation function has obtained the lowest loss of 6.0553e-05, with the highest error percentage of 0.06 MSE, 0.24 RMSE, 0.06 MAE. The next step is to do the same examination with another activation function called Sigmoid, which can be seen in Table 3 shown below.

Table 3. The examination result with ratio of 80:20 using Sigmoid

Look Back	LSTM Units	Epochs	Learning rate	Loss	MSE	RMSE	MAE
15	15	10	0.01	0.0045	0.02 (train) 0.00 (test)	0.13 (train) 0.00 (test)	0.02 (train) 0.00 (test)
			0.001	0.1590	0.39 (train) 0.00 (test)	0.63 (train) 0.07 (test)	0.39 (train) 0.00 (test)
		100	0.01	0.015	0.00 (train) 0.00 (test)	0.05 (train) 0.01(test)	0.00 (train) 0.00 (test)
			0.001	0.0052	0.01 (train) 0.00 (test)	0.11 (train) 0.00 (test)	0.01 (train) 0.00 (test)

Look Back	LSTM Units	Epochs	Learning rate	Loss	MSE	RMSE	MAE
	50	10	0.01	0.0011	0.05 (train) 0.00 (test)	0.22 (train) 0.01 (test)	0.05 (train) 0.00 (test)
			0.001	0.1247	0.29 (train) 0.00 (test)	0.54 (train) 0.07 (test)	0.29 (train) 0.00 (test)
		100	0.01	**9.6793e-04**	0.00 (train) 0.00 (test)	0.05 (train) 0.00(test)	0.00 (train) 0.00 (test)
			0.001	0.0035	0.01 (train) 0.00 (test)	0.10 (train) 0.00(test)	0.01 (train) 0.00 (test)

Table 4. The examination result with ratio of 80:20 using Sigmoid (cont.)

Look Back	LSTM Units	Epochs	Learning rate	Loss	MSE	RMSE	MAE
30	15	10	0.01	0.0018	0.05 (train) 0.00 (test)	0.23 (train) 0.01 (test)	0.05 (train) 0.00 (test)
			0.001	0.1533	0.54 (train) 0.02 (test)	0.73 (train) 0.13 (test)	0.54 (train) 0.02 (test)
		100	0.01	5.1107e-04	0.04 (train) 0.00 (test)	0.20 (train) 0.01 (test)	0.04 (train) 0.00 (test)
			0.001	0.0022	0.05 (train) 0.00 (test)	0.23 (train) 0.01 (test)	0.05 (train) 0.00 (test)
	50	10	0.01	0.0011	0.05 (train) 0.00 (test)	0.21 (train) 0.01 (test)	0.05 (train) 0.00 (test)
			0.001	0.1194	**0.64** (train) 0.02 (test)	**0.80** (train) 0.13 (test)	**0.64** (train) 0.02 (test)
		100	0.01	2.1689e-04	0.03 (train) 0.00 (test)	0.18 (train) 0.00 (test)	0.03 (train) 0.00 (test)
			0.001	8.4554e-04	0.03 (train) 0.00 (test)	0.18 (train) 0.01 (test)	0.03 (train) 0.00 (test)

The Table 3 above explains that using the Sigmoid activation function could possibly obtain a higher error percentage such as an RMSE of 0.80, 0.64 MSE, and 0.64 MAE, while the lowest loss obtained is 9.6793e-04. Therefore, the differentiation of the RNN-LSTM performance result uses a dataset with ratio of 80:20, where 80% of the data is used as training data and 20% is used as testing data. Although, the examination has been accomplished, there are various commodities which will be evaluated. The first evaluation for the differentiation of the RNN-LSTM performance result, indicates that implementing 100 epochs for epoch tuning will get the utmost result in comparison to using 10 epochs. In addition, epoch tuning 100 epochs with the Sigmoid activation function gives the result that yields a practically perfect diagram of the result, shown in the Figure 3.

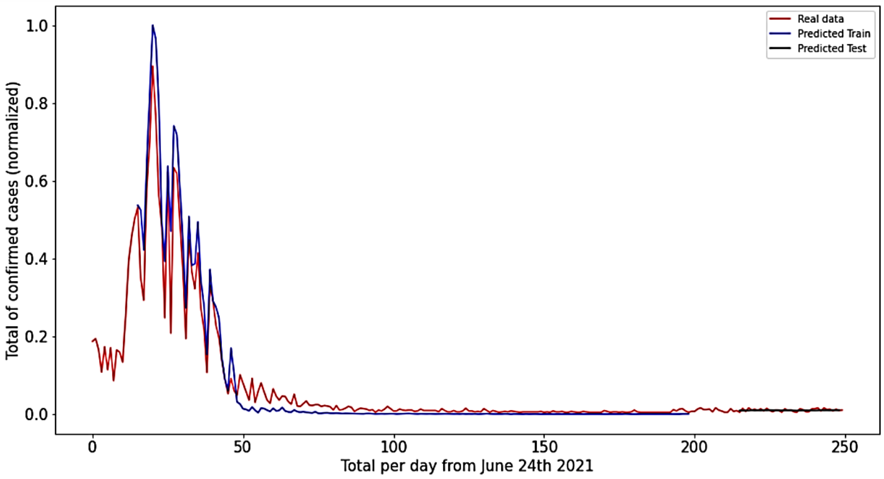

Figure 3. The result of RNN-LSTM

Hence, the second evaluation for the differentiation of RNN-LSTM, associated with the error and loss percentage from the result, indicates that the use of 10 epochs is able to generate an excessive error and loss percentage, and steer the result pattern to differ greatly, far from the dataset pattern. Meanwhile, using 100 epochs results in a minor error and loss percentage, while the pattern of its result does not differ much from the dataset diagram pattern. However, the minor error and loss percentage is resolved by using the Sigmoid activation function, which acquires an RMSE of 0.05, MSE and MAE of 0.03, and a loss of 9.6793e-04, with 100 epochs. Additionally, the enormous error and loss percentage acquired by using the Sigmoid activation function with an RMSE of 0.80, MSE plus MAE of 0.64 and loss of 0.1194, with an epoch tuning of 10 epochs, is shown in the Figure 4 below.

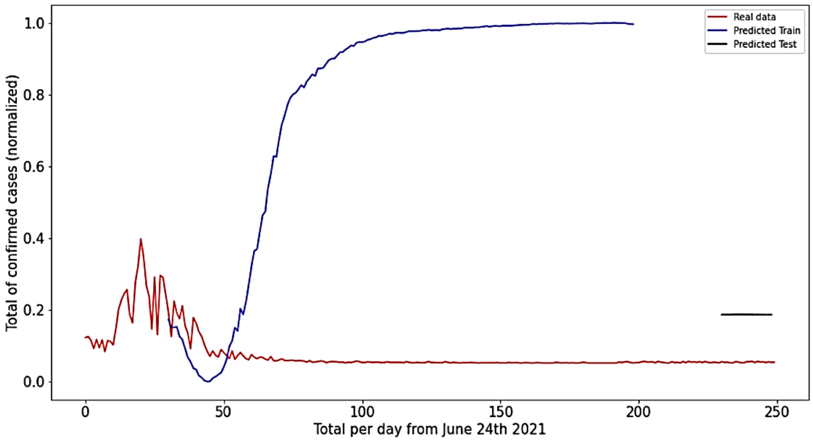

Figure 4. The inadequate result of RNN-LSTM

4. CONCLUSION

Based on the implementations and examinations that have been done, it can be concluded that the implementation of RNN-LSTM in order to predict the Covid-19 patient count in the Tangerang Regency, has been completed. The results of the examination using the Tangerang Regency Covid-19 dataset, which has 250 rows, show a satisfactory result with an error percentage and loss which proved to still be relatively small. Therefore, the best performance is obtained by looking back to the past by 15 days' worth of data, with a ratio of 80:20 and the best epoch tuning, being 100 epochs. In conclusion, the best activation function is the Sigmoid function, while the lowest error percentage or RMSE is 0.05, an MSE of 0.03, and an MAE 0.03, with a loss of 9.6793e-04 using an epoch tuning of 100 epochs.

ACKNOWLEDGEMENT

Authors would like thank you to Universitas Multimedia Nusantara for supporting this research.

REFERENCES

[1] WHO, "Coronavirus disease 2019 (COVID-19) Situation Report – 94," 30 April 2020. [Online]. Available: https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200423-sitrep-94-covid-19.pdf. [Accessed 15 October 2021].

[2] M. Nabila, "Tingkat Keterisian RS Melonjak, Begini Strategi Pemkab Tangerang," jakarta.bisnis.com, 22 January 2021. [Online]. Available: https://jakarta.bisnis.com/read/20210122/383/1346512/tingkat-keterisian-rs-melonjak-begini-strategi-pemkab-tangerang. [Accessed 18 October 2021].

[3] Nikita, "Data Scraping Vs Data Crawling: What's the Difference?," Loginworks, 30 July 2020. [Online]. Available: https://www.loginworks.com/blogs/data-scraping-vs-data-crawling/?__cf_chl_managed_tk__=pmd_d321d005d5b0d9f6bb1a1e1162684ff268f0aa9c-1628759298-0-gqNtZGzNAuKjcnBszQoO#What_Is_Data_Crawling.. [Accessed 15 October 2021].

[4] M. Mohri, A. Rostamizadeh and A. Talwalkar, Foundations of Machine Learning, Cambridge, Massachusetts: The MIT Press, 2012.

[5] A. A. Hania, "Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," June 2017. [Online]. Available: https://www.researchgate.net/publication/320395378_Mengenal_Artificial_Intelligence_Machine_Learning_Neural_Network_dan_Deep_Learning. [Accessed 18 October 2021].

[6] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," 28 May 2015. [Online]. Available: https://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf. [Accessed 15 October 2021].

[7] J. W. G. Putra, Pengenalan Konsep Pembelajaran Mesin dan Deep Learning, Tokyo, Jepang: Jan Wira Gotama Putra, 2018.

[8] P. Vadapalli, "CNN vs RNN: Difference Between CNN and RNN," upgrad.com, 25 February 2021. [Online]. Available: https://www.upgrad.com/blog/cnn-vs-rnn/. [Accessed 18 October 2021].

[9] A. A. Rizal and S. Soraya, "Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory," November 2018. [Online]. Available: https://www.researchgate.net/publication/329338683_Multi_Time_Steps_Prediction_dengan_Recurrent_Neural_Network_Long_Short_Term_Memory. [Accessed 18 October 2021].

[10] R. Nallapati, F. Zhai and B. Zhou, "SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents," 14 November 2016. [Online]. Available: https://arxiv.org/abs/1611.04230. [Accessed 18 October 2021].

[11] K. Ivanedra and M. Mustikasari, "Implementasi Metode Reccurrent Neural Network pada Text Summarization dengan Teknik Abstraktif," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),* vol. 6, no. 4, pp. 377-382, 18 January 2019.

[12] E. D. Tarkus, S. R. Sompie and A. Jacobus, "Implementasi Metode Recurrent Neural Network pada Pengklasifikasian Kualitas Telur Puyuh," *Jurnal Teknik Informatika,* vol. 15, no. 2, pp. 137-144, 30 June 2020.

[13] A. F. Achmalia, Walid and Sugiman, "PERAMALAN PENJUALAN SEMEN MENGGUNAKAN BACKPROPAGATION NEURAL NETWORK DAN RECURRENT NEURAL NETWORK," *UNNES Journal of Mathematics* , vol. 9, no. 1, pp. 6-21, 22 June 2020.

[14] T. A. Zuraiyah, D. K. Utami and D. Herlambang, "IMPLEMENTASI CHATBOT PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN RECURRENT NEURAL NETWORK," *Jurnal Ilmiah Teknologi dan Rekayasa,* vol. 24, no. 2, pp. 91-101, 2019.

[15] M. V. Overbeek, A. G. Y. Stephanno, E. E. Surbakti and F. A. T. Tobing, "Covid-19 Prediction in Indonesia Capital Buffer with K Nearest Neighbor Regression," *2021 5th International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang, Indonesia, 2021, pp. 238-243, doi: 10.1109/ICICoS53627.2021.9651749.

[16] A. Tondak, "Introduction to Recurrent Neural Networks (RNN)," k21academy.com, 4 January 2021. [Online]. Available: https://k21academy.com/datascience/machine-learning/recurrent-neural-networks/#Advantages_Of_RNN_s. [Accessed 18 October 2021].

[17] A. Sarangam, "Recurrent neural network: A Complete Guide In 5 Easy Steps," jigsawacademy.com, 16 December 2020. [Online]. Available: https://www.jigsawacademy.com/blogs/ai-ml/recurrent-neural-network/. [Accessed 18 October 2021].

[18] J. Brownlee, "A Gentle Introduction to Long Short-Term Memory Networks by the Experts," machinelearningmastery.com, 07 July 2021. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/?__cf_chl_captcha_tk__=pmd_54a0f93bdd6a815065c2b3cbdc74a68b6fdd509f-1628834413-0-gqNtZGzNA42jcnBszQni.. [Accessed 18 October 2021].

[19] Vera, A. Kusnadi, I. Z. Pane, M. V. Overbeek and S. G. Prasetya, "Face Recognition Accuracy Improving Using Gray Level Co-occurrence Matrix Selection Feature Algorithm," *2023 International Conference on Smart Computing and Application (ICSCA)*, Hail, Saudi Arabia, 2023, pp. 1-6, doi: 10.1109/ICSCA57840.2023.10087414

BIBLIOGRAPHY OF AUTHORS

Sagita Sasmita Wijaya, a freshly graduated student. Has a passion for learning about Artificial Intelligence, especially Machine Learning and Deep Learning.

Marlinda Vasty Overbeek, lecturer in Department of Informatics at Universitas Multimedia Nusantara. An expert in artificial intelligence, machine learning, natural language processing and expert systems.