JTAM (Jurnal Teori dan Aplikasi Matematika)
http://journal.ummat.ac.id/index.php/jtam
p-ISSN 2597-7512 | e-ISSN 2614-1175
Vol. 6, No. 1, January 2022, pp. 10-21

# Prediction Active Case of Covid-19 with ERNN

**Winda Aprianti[1], Jaka Permadi[2], Herfia Rhomadhona[3]**
[1,2,3]Information Technology, Politeknik Negeri Tanah Laut, Indonesia
winda@politala.ac.id[1], jakapermadi.88@politala.ac.id[2], herfia.rhomadhona@politala.ac.id[3]

**Article History:**

Received : 05-10-2021
Revised : 29-11-2021
Accepted : 05-12-2021
Online : 22-01-2022

**Keywords:**

Active Case;
Covid-19 Pandemic;
Neural Network;
ERNN Algorithm;

**ABSTRACT**

SARS-CoV-2 is known as Covid-19 has been spread in all world since end of 2019. Indonesia, including South Kalimantan has detected first Covid-19 in March 2020. This pandemic has affected in all entirely live in Indonesia. This makes Covid-19 be the main focus of the government. The government has provided aid and imposed restrictions on activities. These policies require planning that can be a solution. Careful planning requires an overview of the data on active cases that are positive for Covid-19. This overview can be obtained through prediction. In this research, Elman Recurrent Neural Network (ERNN) was used to predict active cases of Covid-19. Architecture of ERNN was used ERNN with 3 input nodes, 2 hidden nodes, and 2 context nodes. The data used is 277 data, which is then divided into training data and testing data, respectively 90%-10%, 80%-20%, and 70%-30%. ERNN with a learning rate of 0.1 until 0.9 is applied to data on active cases of Covid-19, then Mean Absolute Percentage Error (MAPE) is calculated to find out performance of model generated by ERNN. The results showed that all of MAPE were below 10% with the smallest MAPE as 3.21% for scenario 90:10 and learning rate 0.6. MAPE value which is less than 10% indicates that ERNN has very good predictive ability.

Crossref
https://doi.org/10.31764/jtam.v6i1.4874



———————— ◆ ————————

## A. INTRODUCTION

Everyone in the world has been fighting against SARS-CoV-2 or known as Covid-19 since the last two years, including The Indonesia Government. In South Kalimantan, Covid-19 was found for the first time in March 2020. The Government has made various efforts, such as imposing restrictions on activities and providing vaccines. However, the spread of Covid-19 still continues and tends to increase. The worst possibility can happen if the number of active cases soars but is not matched by the availability of facilities and personnel health. This can be avoided by making an action plan if it happens. And the careful planning requires an overview of the data on active cases that are positive for Covid-19. This overview can be obtained through prediction.

The use of predictions as a problem solution has often been done with various methods and algorithm. Comparison of neural networks with other various time series forecasting methods have been carried out in previous studies. Neural networks compare with Logistic Regression and Adaptive Neuro-Fuzzy Inference System (ANFIS) (Choubin, Khalighi-Sigaroodi, Malekian, & Kişi, 2016), with statistical method, namely ARIMA (Abdul & Ashour, 2018; Pal, Singh, & Dutta, 2015), with Exponential Smoothing (Aprianti, Permadi, & Rhomadhona, 2020). These researches show that neural networks better method than others. Discuss algorithm and technic of data mining in health field (Madadipouya, 2017) mentioned that artificial neural networks can handle nosy and uncertain data. This is supported by (Chen, 2018) which states that neural networks can be used to solve some problems because it has powerful prediction.

Many various of neural networks have been used to predict many fields in previous researches. One of them is Elman Recurrent Neural Network (ERNN). Context layer of ERNN receive input from hidden layer gives advantage ERNN to learn relation between future values and past ones (Krichene, Masmoudi, & Alimi, 2017). ERNN gives an acceptable prediction when apply to predict reactor accident (Vaziri, Erfani, & Nilforooshan, 2012), consumer price index of education, recreation, and sports in Yogyakarta (Wutsqa, Kusumawati, & Subekti, 2014), opening stocks price (Zheng, 2015), normalized difference vegetation index (NDVI) (Stepchenko & Chizhov, 2015), daily data of peak load (Ismael & Ismael, 2019), sentiment of product review (Astuti & Pratika, 2019) and polarity of a restaurant (Widiastuti & Ali, 2021), and wind speed for wind turbine (Dinzi & Energy, 2020). ERNN has also been compared with other neural network methods.

In Economic Field, ERNN has been compared with Multilayer Perceptron (MLP), Radial Basis Function Neural Network (RBFNN), Jordan Neural Network (JRNN) to predict stock price to Bank of Palestine (Altelbany & Abualhussein, 2021), with Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) to predict stock price of IAM.PA and ORA.PA (Berradi, Lazaar, Omara, & Mahboub, 2020). The result of comparing methods in economic data show that ERNN is the best method. In meteorological field, ERNN also has been compared with Feed Forward Neural Network (FFNN), Cascade Forward Neural Network (CFNN), Recurrent Neural Network (RNN) to predict rainfall that need non-linear technique (Dada, Yakubu, & Oyewola, 2021), Back Propagation Network (BP) to predict wind speed (Li, Zhang, & Mao, 2011). In other fields, ERNN has been compared with fully RNN, JRNN, LSTM, and FFNN to predict tourist visit (Sugiartawan & Hartati, 2019), with ANN and Existing Boosted NN on electric load data, namely voltage profile and active power losses (Devarajan & S, 2019). The application of ERNN on Covid-19 data has been used, but data positive Covid-19 to predict diagnosing it early on (Prasetiyo, Hakim, & Pradana, 2021). In those researches, ERNN gives the best performance than other methods. Its performance can be seen from various criteria, namely Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE). If the criteria give inconsistent result, then we would be chosen MAPE because it is more stable than other criteria (Wang, Wang, Fang, & Niu, 2016).

How good ERNN algorithm is in predicting various time series types has also been carried out in previous study. (Khaldi & Afia, 2018) have compare performance of MLP, ERNN, JRNN, and RBFNN on four types time series, namely trend, seasonality, trend with seasonality, and non-constant variance. This study can prove ERNN is the most suited for time series forecasting. In depth studies, ERNN has been applied with differences in maximal epochs, learning rates, and hidden node (Farhana & Sophiayati, 2015; Prasetiyo et al., 2021). Therefore, this study will apply ERNN on active case of Covid-19 with differences in proportion of training and testing data and differences in learning rate. While the performance of ERNN will be seen from MAPE values

## B. METHODS

In this study, we will show how to apply ERNN manually and with a system that we built. Our study starts from collecting data on active case of Covid-19. That is seconder data that we collect from http://atapdata.ai. The second stage is to form dataset. The first step in this stage is to determine the maximum and minimum value that lie outside the data range. This is because predictions will be made and the possibility of predicted value is more than the current maximum value. After that, we calculate normalize the data using Equation (1).

$$x_i' = \frac{x_i - \min_i(x_i)}{\max_i(x_i) - \min_i(x_i)} \quad (1)$$

where $x_i$ is data of active case in record-$i$, $x_i'$ is normalization of data in record-$i$, and $i$ is index of data records. And then we define values 3 and 2 respectively as number of input node and hidden nodes. It effects on value of context node is 2 too. In architecture of Elman Recurrent Neural Network (ERNN), we have input unit as $x_i$, hidden unit as $h_j$, context unit as $c_k$, output unit as $y$, $v_{0j}$ as weight of bias $b_1$, $w_0$ as weight of bias $b_2$ and respectively $v_{ij}, u_{kj}, w_j$ and weight from network between input - hidden unit, context - hidden unit, and hidden - output unit. This architecture can be seen in Figure 1.

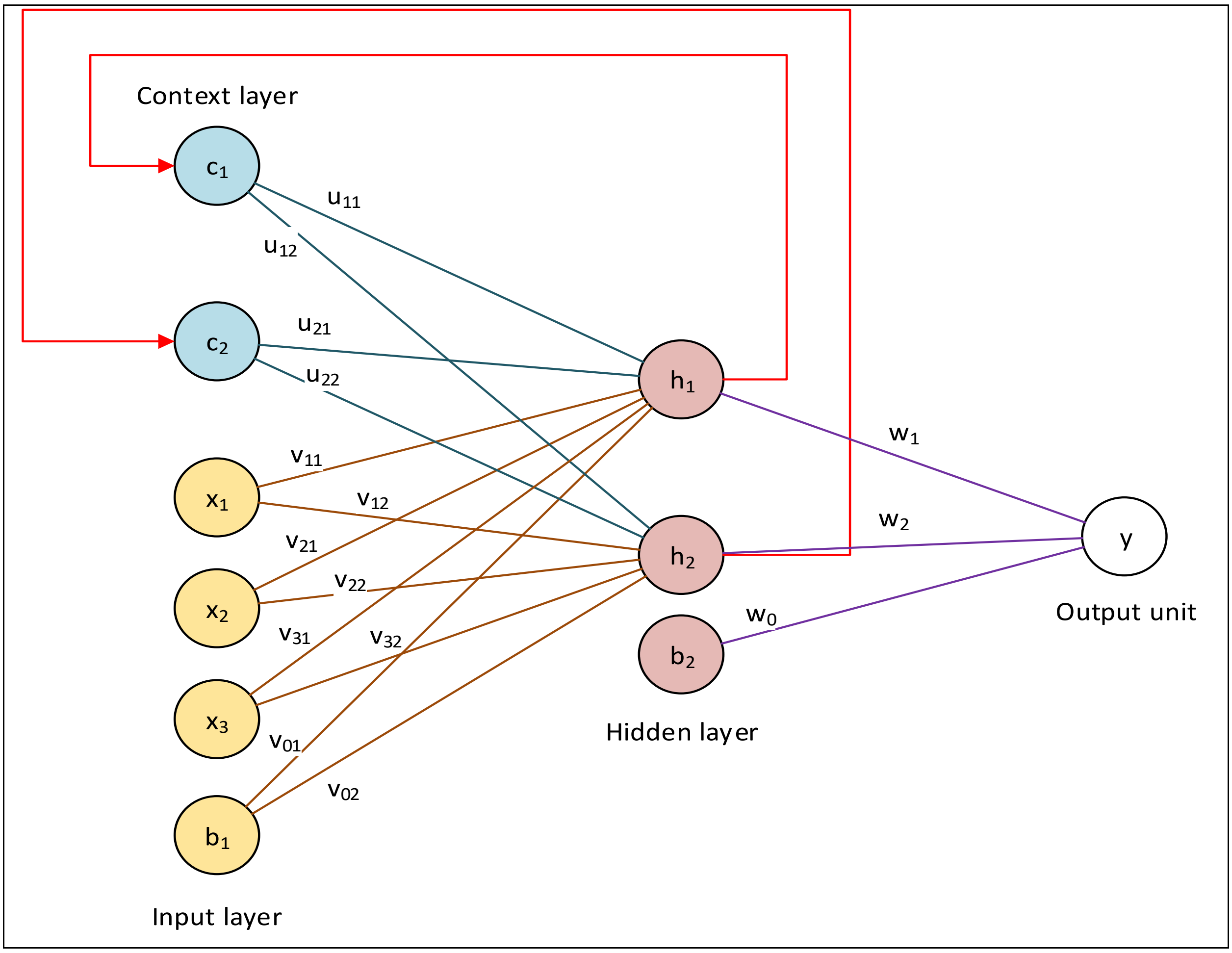


**Figure 1.** Architecture of ERNN

The fourth step in this stage is to determine dataset based on the number of input node. The last step is to divide dataset into training data and testing data. In this research, based on ratio between training and testing data, dataset is design by three scenarios, namely data with ratio 90:10, 80:20, and 70:30. In each scenario, we will use varying learning rate $\alpha$ from 0.1 until 0.9 with an increment as 0.1.

The third stage is to apply training algorithm to data training. The goal of this stage to achieve optimal weight of network. The training algorithm can be presented as shown in Figure 2.

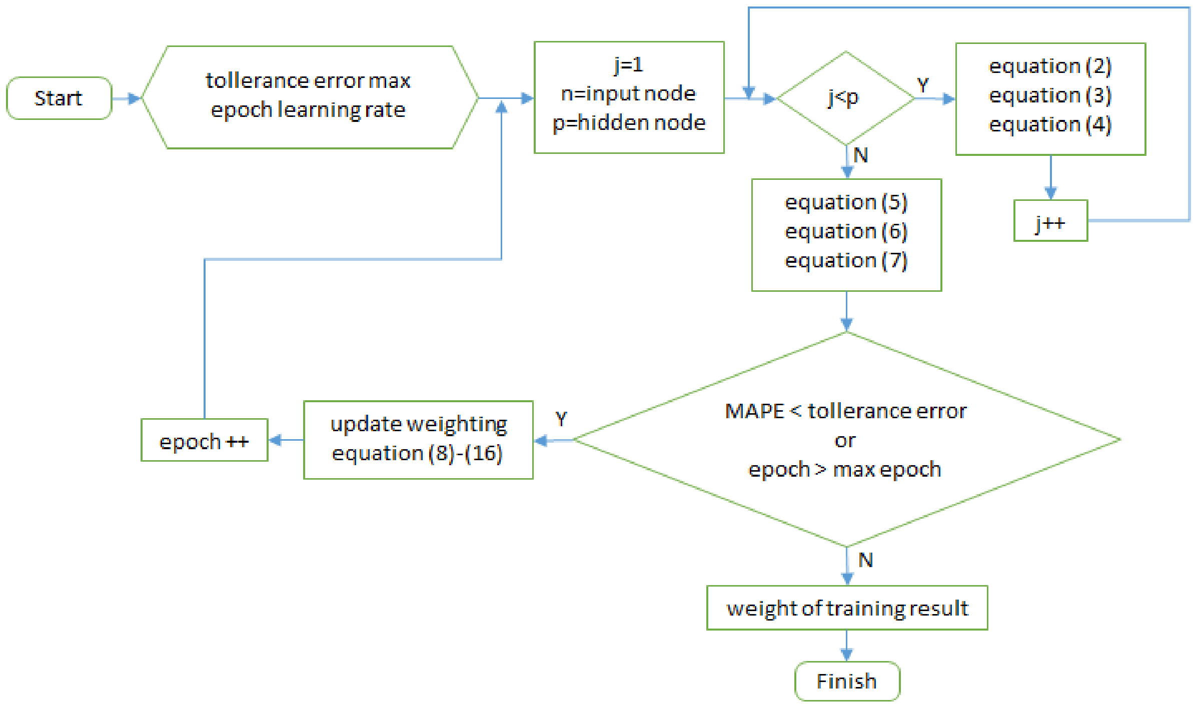


**Figure 2.** Training Algorithm

The following were equation that mentioned in Figure 2.

$$net_j = \sum_{i=1}^{m} x_i v_{ij} + \sum_{k=1}^{n} c_k u_{kj} + v_{oj} \quad (2)$$

$$h_j = f(net_j) = \frac{1}{1+e^{-net_j}} \quad (3)$$

$$c_k(t) = h_k(t-1) \quad (4)$$

$$net_{out} = \sum_{j=1}^{n} h_j w_j + w_0 \quad (5)$$

$$y = f(net\_out) = \frac{1}{1+e^{-net\_out}} \quad (6)$$

$$MAPE = \frac{\sum_{t=1}^{p} \left|\frac{e_t}{d_t}\right|}{p} \times 100\% \quad (7)$$

There is update the weight in Figure 2. To repair the weight, we use Equation (8) until Equation (16).

$$\delta_{out} = y'(d-y) = y(1-y)(d-y) \quad (8)$$

$$\Delta w_j = \alpha \cdot \delta_{out} \cdot h_j \quad (9)$$

$$\Delta w_0 = \alpha \cdot \delta_{out} \quad (10)$$

$$\delta_j = h_j' \cdot \delta_{out} \cdot w_j = h_j(1-h_j) \cdot \delta_{out} \cdot w_j \quad (11)$$

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \quad (12)$$

$$\Delta v_{0j} = \alpha \cdot \delta_j \quad (13)$$

$$w_j(baru) = w_j(baru) + \Delta w_j \quad (14)$$

$$v_{ij}(baru) = v_{ij}(baru) + \Delta v_{ij} \quad (15)$$

$$u_{kj}(baru) = u_{kj}(baru) + \Delta u_{kj} \quad (16)$$

The fourth stage is to apply testing algorithm to data testing as shown in Figure 3.

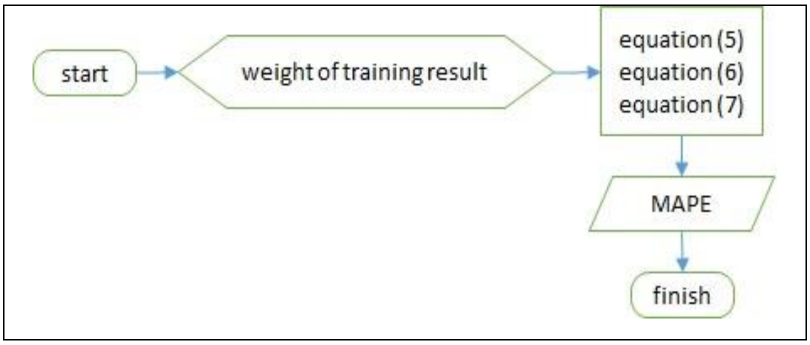


**Figure 3.** Testing Algorithm

## C. RESULT AND DISCUSSION

### 1. Data and Dataset

Data active case of Covid-19 in South Kalimantan is collected from March 30 until December 31, 2020 so it consists of 277 records. According to explanation in method, we need data example to apply ERNN manually, so we have chosen 11 records from December 21 until December 31 as data example. These data as shown in Table 1.

**Table 1.** Data Example

| Number | ID | Date | Month | Active Case |
|---|---|---|---|---|
| 1 | 1221 | 21 | 12 | 858 |
| 2 | 1222 | 22 | 12 | 888 |
| 3 | 1223 | 23 | 12 | 937 |
| 4 | 1224 | 24 | 12 | 996 |
| 5 | 1225 | 25 | 12 | 1014 |
| 6 | 1226 | 26 | 12 | 984 |
| 7 | 1227 | 27 | 12 | 992 |
| 8 | 1228 | 28 | 12 | 973 |
| 9 | 1229 | 29 | 12 | 931 |
| 10 | 1230 | 30 | 12 | 931 |
| 11 | 1231 | 31 | 12 | 911 |

Then we normalize data in Table 1 using Equation (1). As example, we calculate data active case on December 21, namely $x_1$. We determine maximum as 1050 and minimum as 800.

$$x_1' = \frac{x_1 - \min_i(x_i)}{\max_i(x_i) - \min_i(x_i)} = \frac{858 - 800}{1050 - 800} = 0.2320$$

With the same way, we got data normalization as shown in Table 2.

**Table 2.** Data Normalization

| Number | ID | Normalize |
|---|---|---|
| 1 | 1221 | 0.2320 |
| 2 | 1222 | 0.3520 |
| 3 | 1223 | 0.5480 |
| 4 | 1224 | 0.7840 |
| 5 | 1225 | 0.8560 |
| 6 | 1226 | 0.7360 |
| 7 | 1227 | 0.7680 |
| 8 | 1228 | 0.6920 |
| 9 | 1229 | 0.5240 |
| 10 | 1230 | 0.5240 |
| 11 | 1231 | 0.4440 |

Based on data in Table 2 and the number of input node is 3, we get dataset in Table 3. In order to distinguish the dataset used manual and system, we call dataset in Table 3 as Dataset 1.

**Table 3.** Dataset 1

| Date of Prediction (t) | Input | | | Target (d) | Output (y) |
|---|---|---|---|---|---|
| | $x_1$ (t-3) | $x_2$ (t-2) | $x_3$ (t-1) | | |
| 1224 | 0.2320 | 0.2933 | 0.4567 | 0.7840 | 0.2320 |
| 1225 | 0.3520 | 0.4567 | 0.7840 | 0.8560 | 0.3520 |
| 1226 | 0.5480 | 0.7840 | 0.8560 | 0.7360 | 0.5480 |
| 1227 | 0.7840 | 0.8560 | 0.7360 | 0.7680 | 0.7840 |
| 1228 | 0.8560 | 0.7360 | 0.7680 | 0.6920 | 0.8560 |
| 1229 | 0.7360 | 0.7680 | 0.6920 | 0.5240 | 0.7360 |
| 1230 | 0.7680 | 0.6920 | 0.5240 | 0.5240 | 0.7680 |
| 1231 | 0.6920 | 0.5240 | 0.5240 | 0.4440 | 0.6920 |

Application of ERNN on Dataset 1 would be found in C.2. Now, we will form Dataset 2. Regard to data that we have start from March 30, 2020 and number of input node is 3, then data of active case on March 30, 31, and April 1 would be input value to predict active case on April 2. Do the same way until data of active case on December 31, 2020 so we get the Dataset 2 that consist 274 records. Application of ERNN on Dataset 2 would be found in C.3

## 2. Apply ERNN Manually

In this stage, we determined learning rate α as 0.2, error tolerance as 0.01 or 1%, and maximal epochs (max epoch) as 3. The following are calculation example in first epoch. The value of context unit 1 is 0 because there was no context unit in date before. Then the initial weight is chosen randomly as shown in Table 4.

**Table 4.** Initial Weight

| Between | | Weight |
|---|---|---|
| Input unit - Hidden Unit ($v_{ij}$) | $v_{01}$ | 0.6113 |
| | $v_{02}$ | 0.3180 |
| | $v_{11}$ | 0.3918 |
| | $v_{12}$ | 0.9933 |
| | $v_{21}$ | 0.0158 |
| | $v_{22}$ | 0.0840 |
| | $v_{31}$ | 0.2700 |
| | $v_{32}$ | 0.2019 |
| Context Unit - Hidden Unit ($u_{kj}$) | $u_{11}$ | 0.3289 |
| | $u_{12}$ | 0.2530 |
| | $u_{21}$ | 0.8089 |
| | $u_{22}$ | 0.6266 |
| Hidden Unit - Output Unit ($w_j$) | $w_0$ | 0.3223 |
| | $w_1$ | 0.2019 |
| | $w_2$ | 0.0274 |

The following are calculation of hidden unit to 1 ($h_1$) and its illustration as shown in Figure 4.

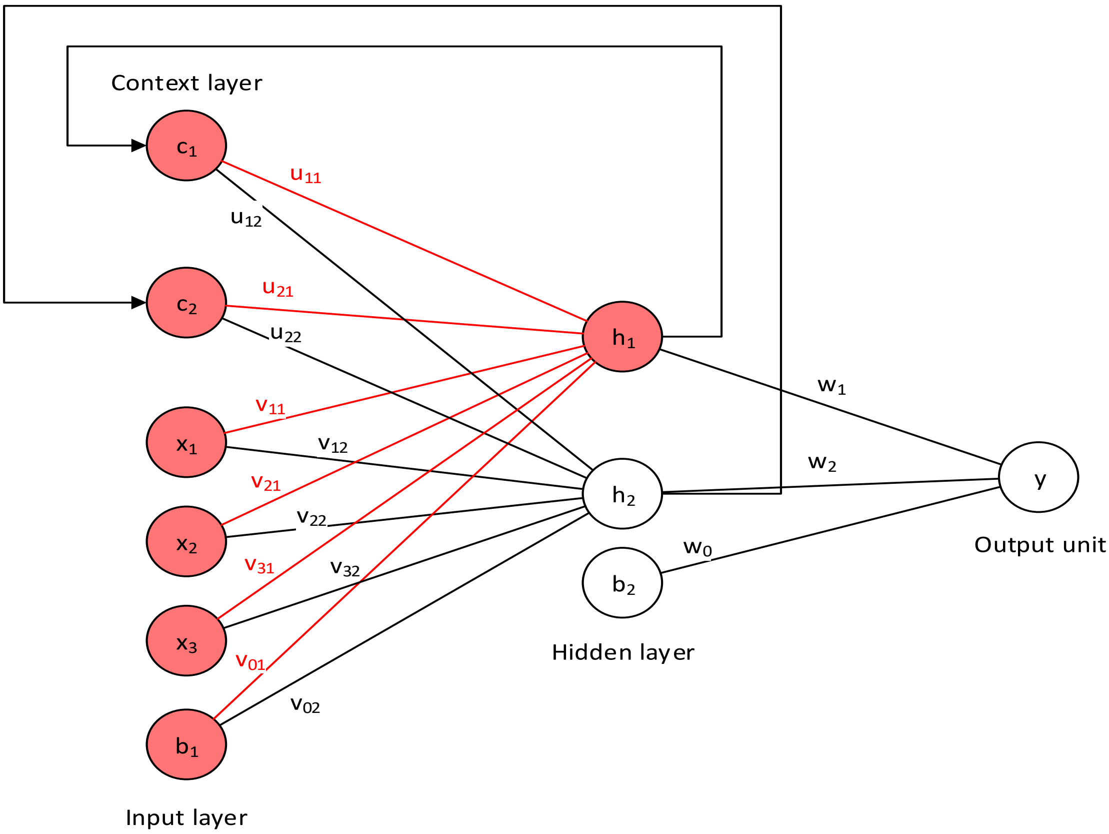


**Figure 4.** Illustration of $h_1$ Calculation

We calculate $net_1$ of data 1 using Equation (2) and $h_1$ using Equation (3).

$$net_1 = \sum_{i=1}^{3} x_i v_{i1} + \sum_{k=1}^{2} c_k u_{k1} + v_{01} = x_1 v_{11} + x_{22} v_{21} + x_3 v_{31} + c_1 u_{11} + c_2 u_{21} + v_{01}$$

$$= 0.232 \times 0.3918 + 0.2933 \times 0.0158 + 0.4567 \times 0.27 + 0 \times 0.3289$$
$$+0 \times 0.80889 + 0.6113$$
$$= 0.8301$$

$$h_1 = f(net_1) = \frac{1}{1 + e^{-net_1}} = 0.6964$$

With the same way to calculate $h_1$, value of $h_2$ is got, that is 0.6604. After value of $h_1$ and $h_2$ is got, then context unit can be calculated by using Equation (4).

$$c_1(2) = h_1(1) = 0.6964$$
$$c_2(2) = h_2(1) = 0.6604$$

Then $y$ is calculated by using Equation (5) and (6). Its illustration as shown in Figure 5.

$$net_{out} = \sum_{j=1}^{2} h_j w_j + w_0 = h_1 w_1 + h_2 w_2 + w_0$$
$$= 0.6964 \cdot 0.2019 + 0.6604 \cdot 0.0274 + 0.3223 = 0.481$$
$$y = f(net\_out) = f(0.481) = \frac{1}{1 + e^{-0.481}} = 0.618$$

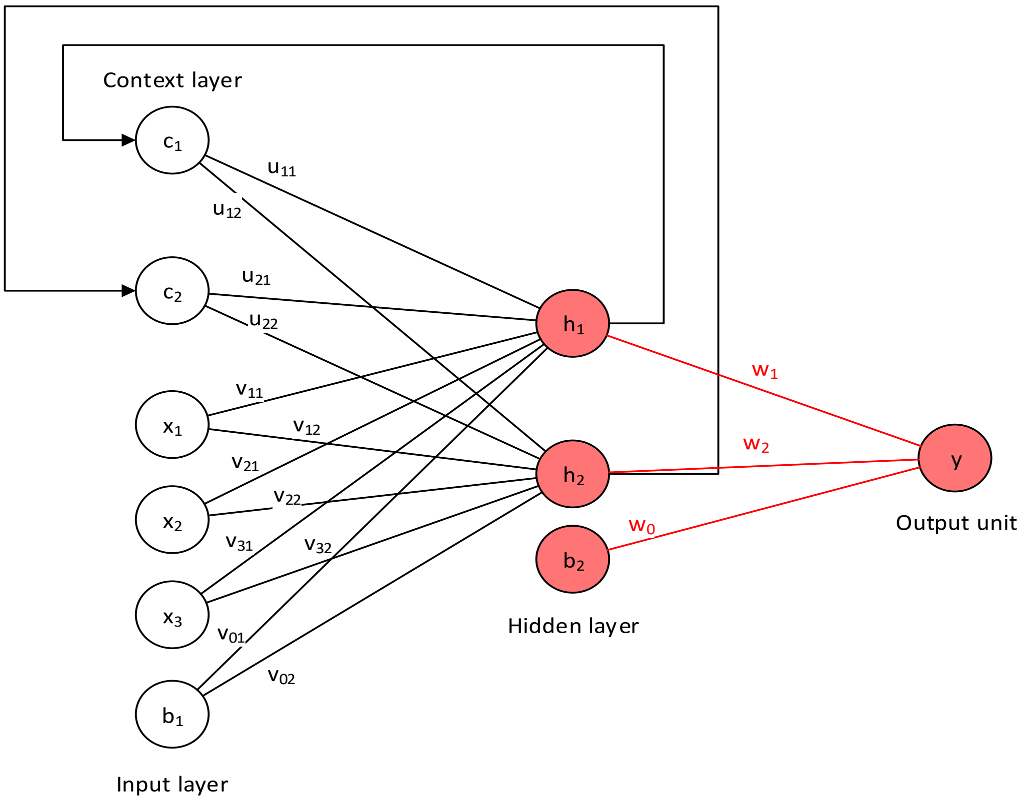


**Figure 5.** Illustration of $y$ Calculation

The rest data in training data is calculated with the same way and the result as shown in Table 5.

**Table 5**. The result of Epoch 1

| Date of prediction(t) | Input | | | Context | | Hidden | | Target (d) | Output (y) |
|---|---|---|---|---|---|---|---|---|---|
| | x1 (t - 3) | x2 (t - 2) | x3 (t - 1) | c1 | c2 | h1 | h2 | | |
| 1224 | 0.232 | 0.2933 | 0.4567 | 0 | 0 | 0.6964 | 0.6604 | 0.7840 | 0.618 |
| 1225 | 0.352 | 0.4567 | 0.784 | 0.6964 | 0.6604 | 0.8496 | 0.8107 | 0.8560 | 0.6262 |
| 1226 | 0.548 | 0.784 | 0.856 | 0.8496 | 0.8107 | 0.8813 | 0.861 | 0.7360 | 0.628 |
| 1227 | 0.784 | 0.856 | 0.736 | 0.8813 | 0.861 | 0.8926 | 0.8889 | 0.7680 | 0.6288 |
| 1228 | 0.856 | 0.736 | 0.768 | 0.8926 | 0.8889 | 0.8983 | 0.8973 | 0.6920 | 0.6291 |
| 1229 | 0.736 | 0.768 | 0.692 | 0.8983 | 0.8973 | 0.8928 | 0.8852 | 0.5240 | 0.6287 |

Based on Table 5 MAPE is calculated using Equation (7).

$$MAPE = \frac{\sum_{t=1}^{6} \left|\frac{e_t}{d_t}\right|}{6} \times 100\% = \frac{1.0989}{6} \times 100\% = 18.32\%$$

Thus $MAPE > tollerance\ error$ and $epoch < maxepoch$ then we updating weight using Equation (8) until Equation (16). Update weight of training result in epoch 1 is used as weight in epoch 2 and repeat the steps in epoch 1. But the both of MAPE in epoch 2 and epoch 3 bigger than tolerance error, so we updating weight in epoch 3and then it uses to predict testing data. The result as shown in Table 6.

**Table 6**. The result of Epoch 1

| Date of prediction (t) | Input | | | Context | | Hidden | | Target (d) | Output (y) |
|---|---|---|---|---|---|---|---|---|---|
| | x1 (t - 3) | x2 (t - 2) | x3 (t - 1) | c1 | c2 | h1 | h2 | | |
| 1230 | 0.5434 | 0.6551 | 0.6095 | 0.9025 | 0.8856 | 0.8922 | 0.8606 | 0.7090 | 0.6639 |
| 1231 | 0.6551 | 0.6095 | 0.7090 | 0.8922 | 0.8606 | 0.8965 | 0.8731 | 0.8046 | 0.6644 |

Based on Table 6, we calculate MAPE using Equation 7 and we get MAPE as 3.97%. This result indicates that ERNN is the good method to predict active case in Dataset 1.

## 3. Apply ERNN with System

Through the system was built, we apply ERNN alternately for 90:10, 80:20, and 70:30. In scenario 90:10, input proportion of training as 90% and testing data as 10%, those are 247 and 27 records. The scenario 80:20 given data as 219 and 55 records, and scenario 70:30 given data as 192 and 82 records. Then we set learning rate $\alpha$ from 0.1 until 0.9 with an increment as 0.1, maximum epoch as 100.000, and tolerance error as 0.000001. The result of applying ERNN with this system as shown in Figure 6.

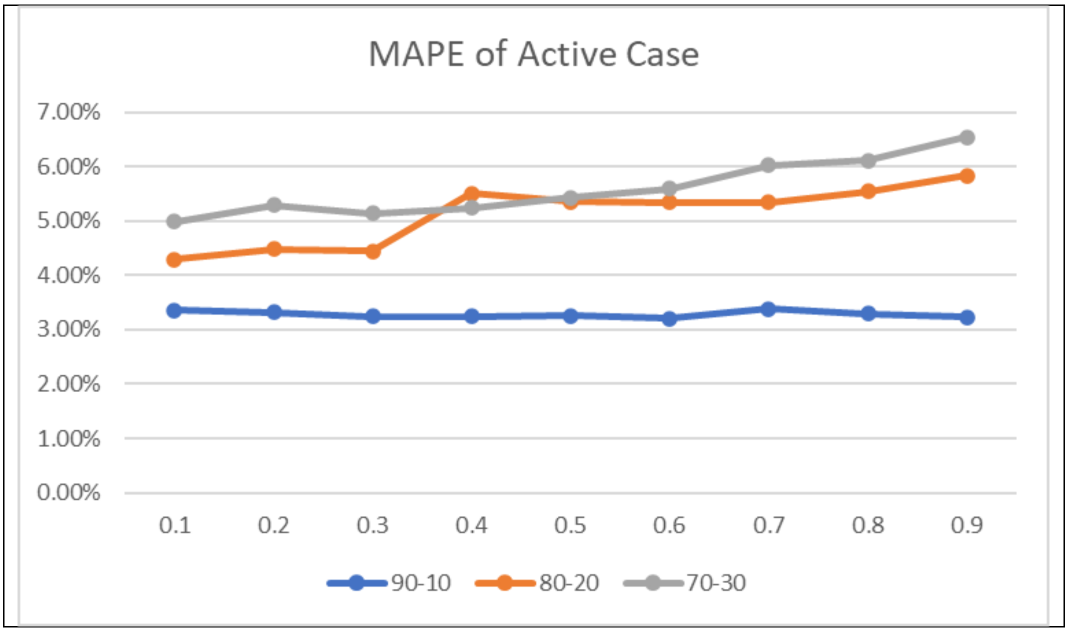


**Figure 6.** The Result of Apply ERNN with System

In overall from Figure 6, all MAPE below 10% and the smallest MAPE is scenario 90:10. In scenario 90:10, the smallest MAPE as 3.21% for learning rate $\alpha = 0.6$, meanwhile in scenario 80:20 and 70:30, the smallest MAPE for learning rate $\alpha = 0.1$. The results show that ERNN has been well implemented on COVID-19 data in accordance with research results (Khaldi & Afia, 2018) that ERNN is suitable for predicting time series data. The success of ERNN in predicting fluctuating Covid-19 data is also in accordance with research to predict fluctuating data on the opening of stock prices by (Zheng, 2015).

## D. CONCLUSION AND SUGGESTIONS

ERNN has been applied on active case of Covid-19, namely Dataset 1 and Dataset 2. Dataset 1 was divided into 90% training data and 10% training data, set learning rate as 0.2, tolerance error as 0.01, and maximal epoch as 3. Application of ERNN on Dataset 1 gave MAPE as 3.97%. Meanwhile, 3 scenarios on Dataset 2 were ratio 90:10, 80:20, and 70:30. In each ratio, we set learning rate from 0.1 until 0.9 with 0.1 increments, maximum epoch as 100.000, and tolerance error as 0.000001. Application of ERNN on Dataset 2 provides all MAPE below 10% with the smallest MAPE as 3.21% for scenario 90:10 and learning rate 0.6. This shows that ERNN is a good method to predict active case of Covid-19. However, the accuracy can be classified as quite high because this research relates to human safety. Therefore, an in-depth study on predict of active case is needed. I hope that future research can compare ERNN with other methods, neural networks algorithms or another, to find the best method to predict active case of Covid-19. Researchers interested in this study, can also use difference increments of learning rate to test how well ERNN predicts.


## ACKNOWLEDGEMENT

This research is a part of a project financed by DIPA Politeknik Negeri Tanah Laut in Penelitian Dosen Dana DIPA scheme (009/PL40.5/LT/2021) fiscal year 2021.


## REFERENCES


Abdul, M., & Ashour, H. (2018). Improving Time Series 'Forecast Errors by Using Recurrent Neural Networks. *ICSCA 2018*, (February), 229–232. https://doi.org/10.1145/3185089.3185151

Altelbany, S. I., & Abualhussein, A. A. (2021). Performance Comparison of Neural Networks ( MLP , RBFNN , ERNN , JRNN ) Models for Stock Prices Forecasting to Bank of Palestine. *Muthanna Journal of Administrative and Economic Sciences*, *11*(1), 8–28. https://doi.org/10.52113/6/2021-11/8-28

Aprianti, W., Permadi, J., & Rhomadhona, H. (2020). Perbandingan Elman Recurrent Neural Networks, Backpropagation Neural Networks, dan Exponential Smoothing dalam Peramalan Produksi Palawija. *MUST: Journal of Mathematics Education, Science and Technology*, *5*(2), 206–220. https://doi.org/http://doi.org/10.30651/must.v5i2.6255

Astuti, T., & Pratika, I. (2019). Product Review Sentiment Analysis by Artificial Neural Network Algorithm. *International Journal of Informatics and Information Systems*, *2*(2), 61–66.

Berradi, Z., Lazaar, M., Omara, H., & Mahboub, O. (2020). Effect of Architecture in Recurrent Neural Network Applied on the Prediction of Stock Price. *IAENG International Journal of Computer Science*, *47*(3), 199–204.

Chen, Y. (2018). Applications of Recurrent Neural Networks in Environmental Factor Forecasting : A Review. *Neural Computation*, *30*, 2855–2881. https://doi.org/10.1162/neco_a_01134

Choubin, B., Khalighi-Sigaroodi, S., Malekian, A., & Kişi, Ö. (2016). Multiple linear regression, multi-layer perceptron network and adaptive neuro-fuzzy inference system for forecasting precipitation based on large-scale climate signals. *Hydrological Sciences Journal*, *61*(6), 1001–1009. https://doi.org/10.1080/02626667.2014.966721

Dada, E. G., Yakubu, H. J., & Oyewola, D. O. (2021). Artificial Neural Network Models for Rainfall Prediction. *EJECE, European Journal of Electrical Engineering and Computer Science*, *5*(2), 30–35. https://doi.org/http://dx.doi.org/10.24018/ejece.2021.5.2.313

Devarajan, S., & S, C. (2019). Load Forecasting Model for Energy Management System Using Elman Neural Network. *International Research Journal of Multidisciplinary Technovation (IRJMT)*, *1*(3), 48–56. https://doi.org/https://doi.org/10.34256/irjmt1936

Dinzi, R., & Energy, A. W. (2020). The Use of Meteorology Data in Short-Term Prediction of Wind Speed for Wind Turbine Using Elman Recurrent Neural Network. *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 93–98.

Farhana, N., & Sophiayati, S. (2015). Forecasting Rainfall Distribution Using Artificial Neural Networks for Johor Rivers. *Open International Journal of Informatics (OIJI)*, *3*(1), 11–27.

Ismael, L., & Ismael, A. I. (2019). Prediction the data consumption for power demands by Elman neural network. *International Journal of Electrical and Computer Engineering (IJECE)*, *9*(5), 4003–4009. https://doi.org/10.11591/ijece.v9i5.pp4003-4009

Khaldi, R., & Afia, A. El. (2018). Feedforward and Recurrent Neural Networks for Time Series Forecasting : Comparative Study. *Proceedings of ACM LOPAL Conference*, 1–6. https://doi.org/https://doi.org/10.1145/3230905.3230946

Krichene, E., Masmoudi, Y., & Alimi, A. M. (2017). Forecasting Using Elman Recurrent Neural Network. *Intelligent Systems Design and Applications*, *2*(ii), 488–497. https://doi.org/10.1007/978-3-319-53480-0

Li, J., Zhang, B., & Mao, C. (2011). Wind speed prediction and error distribution based on rational sample organisation for Elman recursion neural networks. *International Journal Advanced Mechatronic Systems*, *3*(July 2010), 17–19.

Madadipouya, K. (2017). A survey on data mining algorithms and techniques in medicine. *International Journal on Informatics Visualization*, *1*(3), 61–71. https://doi.org/10.30630/joiv.1.3.25

Pal, A., Singh, J. P., & Dutta, P. (2015). Path length prediction in MANET under AODV routing: Comparative analysis of ARIMA and MLP model. *Egyptian Informatics Journal*, *16*(1), 103–111. https://doi.org/10.1016/j.eij.2015.01.001

Prasetiyo, B., Hakim, M. F. Al, & Pradana, F. D. (2021). Prediction of COVID-19 Using Recurrent Neural Network Model. *Scientific Journal of Informatics*, *8*(1), 98–103. https://doi.org/10.15294/sji.v8i1.30070

Stepchenko, A., & Chizhov, J. (2015). NDVI Short-Term Forecasting Using Recurrent Neural Networks. *Proceedings of the 10th International Scientific and Practical Conference*, *3*, 180–185. https://doi.org/http://dx.doi.org/10.17770/etr2015vol3.167

Sugiartawan, P., & Hartati, S. (2019). Time Series Data Prediction using Elman Recurrent Neural Network on Tourist Visits in Tanah Lot Tourism Object. *International Journal of Engineering and Advanced Technology (IJEAT)*, (1), 314–320. https://doi.org/10.35940/ijeat.A1833.109119

Vaziri, N., Erfani, A., & Nilforooshan, B. (2012). Signal Prediction in the LOCA Using Elman Recurrent Neural Networks. *International Journal of Science and Engineering Investigations*, *1*(7), 1–4.

Wang, J., Wang, J., Fang, W., & Niu, H. (2016). Financial Time Series Prediction Using Elman Recurrent Random Neural Networks. *Computational Intelligence and Neuroscience*, *2016*, 1–14.

Widiastuti, N. I., & Ali, M. I. (2021). Elman Recurrent Neural Network For Aspect Based Sentiment Analysis. *Journal of Engineering Science and Technology*, *16*(3), 1991–2000.

Wutsqa, D. U., Kusumawati, R., & Subekti, R. (2014). The Application of Elman Recurrent Neural Network Model for Forecasting Consumer Price Index of Education, Recreation and Sports in Yogyakarta. *2014 10th International Conference on Natural Computation*, 192–196.

Zheng, J. (2015). Forecast of Opening Stock Price Based on Elman Neural Network. *The Chemical Engineering Transaction*, *46*(2014), 565–570. https://doi.org/10.3303/CET1546095