International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 2
ISSN: 2088-5334

Forecasting the Currency Rate of The Indonesian Rupiah (IDR) against the US Dollar (USD) Using Time Series Data and Indonesian Fundamental Data

Lady Silk Moonlight [a,*], Bambang Bagus Harianto [a], Yuyun Suprapto [a], Fiqqih Faizah [a]

[a] *Politeknik Penerbangan Surabaya, Jemur Andayani 1 No 73, Surabaya, 60236, Indonesia*
*Corresponding author: *lady@poltekbangsby.ac.id*

***Abstract*— In forecasting foreign currencies, or known as foreign exchange, fundamental analysis and technical analysis can be used. Fundamental analysis relies on external factors or news happening in the market. In comparison, technical analysis studies the price itself by relying on graphs and mathematical formulas. This study combines fundamental value and technical analysis to predict the Rupiah (IDR) against the Dollar (USD). In this study, the artificial neural network architecture that is compared is Backpropagation and Recurrent Neural Networks (RNN). The RNN architecture used in this research is Elman and Jordan with Backpropagation Through Time (BPTT) learning algorithm. Technical analysis is applied by entering the USD exchange rate against IDR at a certain time. At the same time, fundamental analysis is applied in the form of entering some data on the value of fundamental factors as a training data set. Fundamental data used in this research are inflation rate, interest rate, money supply, and the number of exports and imports in Rupiah. In this study, the prediction system is also compared, the prediction system uses technical data, and the prediction system uses technical and fundamental data. This research results in the prediction system using the Elman RNN algorithm, which is better than the Backpropagation and Jordan RNN algorithms. A prediction system using training data in time series and fundamental data is better than only training data. So, it means in this study that the best prediction system uses the Elman RNN Algorithm with training data in the form of time series data USD sell exchange rate against IDR and fundamental data.**

***Keywords*— Forecasting rupiah value; fundamental analysis; technical analysis; backpropagation; backpropagation through time; recurrent neural networks.**

Manuscript received 18 Jun. 2022; revised 7 Nov. 2022; accepted 24 Jan. 2023. Date of publication 30 Apr. 2023.

I. INTRODUCTION

Changes in currency exchange rates occur all the time in the world, including the Indonesian Rupiah against the US Dollar. In analyzing changes in currency exchange rates, fundamental and technical analysis techniques are used. Fundamental analysis concentrates on the basic causes of price movements and relies on news/rumors in the market. While technical analysis studies the movement of the price by relying on chart changes using mathematical formulas. Fundamental factors are changes in currency values influenced by inflation rates, interest rates, income levels, currency supply, demand, balance of payments (BOP) positions, government supervision, and speculation expectations. This volatile currency exchange rate movement makes it important to develop the science of forecasting in estimating foreign exchange rates with low risk. The Artificial Neural Network approach, which has been further developed into Deep Learning, has proven to be very effective in performing pattern recognition [1].

In this study, we tried to combine fundamental analysis techniques and technical analysis in predicting the exchange rate of the Indonesian Rupiah against the US Dollar. Application of Artificial Intelligence Recurrent Neural Networks (RNN) and Backpropagation methods as a form of technical analysis. And fundamental analysis is applied in the form of entering data on fundamental factors as a training dataset. In previous research, it was stated that including monthly monetary fundamental volatility, it could improve daily forecasting performance [2]. Fundamental factors and price factors complement each other in forecasting future returns and combining price reversal and fundamental momentum into enhanced fundamental momentum results in a 2.107% monthly return [3]. Previous studies stated that economic fundamentals were proven to improve the forecasting performance of the euro-dollar exchange rate in the time range from one week to one month [4]. By combining the two techniques of technical and fundamental analysis, it is hoped that this research will obtain better results that are not only based on past prices but also take into account other indicators.

From the results of the analysis of articles from 2000 to 2019 about forecasting the index, stock market, and currency values, it is known that 138 articles reveal that machine learning is the most widely used method. Then from 2020 until now, deep learning has been increasingly popular [5]. Machine learning also predicts the daily value of stocks, and currency exchange rates, to the value of commodities. [6] [7] [8]

The Backpropagation method is one of the Artificial Neural Network methods included in Deep Machine learning. In previous research, the Backpropagation Method can handle unexpected spikes or decreases in stock prices due to stock splits and commodity prices approaching the contract's expiration date[9]. The feedforward backpropagation neural network model can accurately model indoor air quality in naturally ventilated buildings, compared to the multivariate statistical method, Multiple Linear Regression (MLR) [10]. Backpropagation Neural Networks can analyze environmental risks and provide fast predictions, with epochs of 2.5E+04, an average RMSE of 1.06E-06, and the average comparison between the measured and predicted outputs is 0.9994 [11]. The Backpropagation Neural Network (BPNN) method has also been compared with the traditional artificial neural network (ANN) in the test set in predicting the risk in The digital transformation of China's manufacturing, which results that BPMN is better than the traditional artificial neural network (ANN) [12].

Recurrent Neural Networks (RNN) can handle very complex financial systems with various time variations [13]. In the case of estimating forest energy, a simulation system built using RNN in proposing optimization schemes and policy recommendations results in a system that can achieve the benefits of reducing carbon dioxide emissions by up to 46.17% and increasing the economic benefits of trading carbon dioxide by 35.69% [14]. In the case of predicting the evolution of the risk of death from COVID-19, the performance of the RNN outperforms the Support Vector Classifier and Random Forest [15]. RNN with Backpropagation through Time (BPTT) algorithm has a powerful computational framework for learning and forecasting complex spatiotemporal systems [16]. RNN, as one of machine learning, has succeeded in optimizing the non-linear process of estimating operational data with accuracy according to estimates [17], [18]. RNN framework with adaptive training strategy is used to model non-linear data dynamic systems for long-term future state prediction [19]. Compared to exponential smoothing (ETS) and the autoregressive integrated moving average (ARIMA), the RNN model shows a higher forecasting accuracy value, is more efficient, with homogeneous seasonal pattern data, and can be a competitive alternative in many situations [20]. Elman Neural Network with direct input-to-output connections has been proven to increase accuracy, and reduce network complexity and computational load, for classification problems or linear data regression, but for non-linear data, this method is not appropriate [21]. The Elman Neural Network model can predict the actual load curve and release time of the energy storage tank storage with a large suitability value to effectively reduce energy use and operating costs without including thermal comfort [22].

In this study, three different architectures were compared: Backpropagation, Elman RNN, and Jordan RNN. From the explanation above, it is known that each architecture has its own advantages. The difference between these architectures is the iterative connection that we have described in section II, Materials and Method. However, the three architectures use the same algorithm steps, namely feedforward and backward, but adapted to each network architecture. The feedforward and backward algorithm steps applied to the RNN architecture are called Backpropagation Through Time (BPTT).

II. Materials and Method

This section describes the steps of the Backpropagation algorithm, which is applied to the network architecture of the Neural Network and the Recurrent Neural Network.

A. *Backpropagation*

The Backpropagation algorithm is used in a multi-layer feedforward artificial neural network composed of several layers, and the signal flows in the same direction from the input to the output layer. The backpropagation training algorithm consists of three stages, namely:

- Input the value of the training data so that the output value is obtained.
- Backpropagation of the obtained error value
- Connection weight adjustment to minimize error value.

The three stages are repeated continuously until the desired error value is obtained. After the training is completed, only the first stage is needed to utilize the artificial neural network. Error information is propagated sequentially, starting from the output layer and ending at the input layer, so this algorithm is called Backpropagation [23], [24], [25], [26], [27], [28].

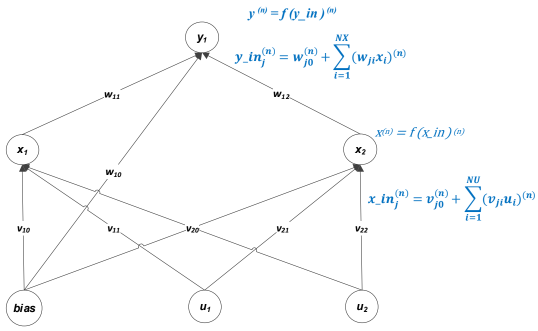

Fig. 1 Feedforward steps of Backpropagation architecture

Feedforward steps of the Backpropagation Algorithm as shown in Fig. 1:

1. Calculate all incoming signals in the *hidden neurons* using equation (1).

$$x\_in_j^{(n)} = v_{j0}^{(n)} + \sum_{i=1}^{NU} (v_{ji}u_i)^{(n)} \tag{1}$$

2. Calculate the output signals of the *hidden neurons* using equation (2).

$$x(n) = f(x\_in)(n) \tag{2}$$

3. Count all incoming signals to the *output neurons* using equation (3).

$$y\_in_j^{(n)} = w_{j0}^{(n)} + \sum_{i=1}^{NX} (w_{ji} x_i)^{(n)} \tag{3}$$

4. Calculate output signals of the *output neurons* using equation (4).

$$y(n) = f(y\_in)(n) \tag{4}$$

5. Calculate the *Root Mean Squared Error (RMSE)* using equation (5).

$$RMSE = \sqrt{\frac{1}{n}\sum (F_t - A_t)^2} \tag{5}$$

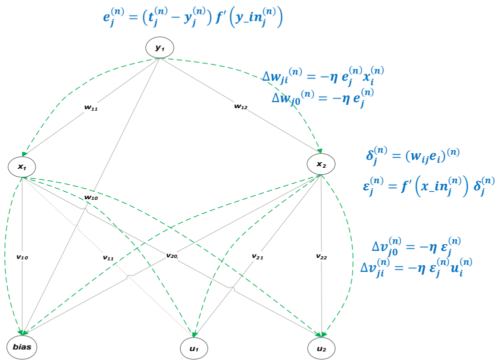

Fig. 2 Backward steps of Backpropagation architecture

Then do the backward steps as shown in Fig. 2 as follows:

6. Calculate the error in the *output units* using equation (6).

$$e_j^{(n)} = \left(t_j^{(n)} - y_j^{(n)}\right) f'\left(y\_in_j^{(n)}\right) \tag{6}$$

7. Calculate weight correction using equations (7) and (8).
Output – Hidden:

$$\Delta w_{ji}{}^{(n)} = -\eta\, e_j^{(n)} x_i^{(n)} \tag{7}$$

Output – Bias:

$$\Delta w_{j0}{}^{(n)} = -\eta\, e_j^{(n)} \tag{8}$$

8. Calculate the sum of the input deltas in the *hidden units* using equation (9).

$$\delta_j^{(n)} = (w_{ij} e_i)^{(n)} \tag{9}$$

Calculate the error in *hidden units* using equation (10).

$$\varepsilon_j^{(n)} = f'\left(x\_in_j^{(n)}\right) \delta_j^{(n)} \tag{10}$$

9. Calculate weight correction using (11) and (12)
Hidden – Input:

$$\Delta v_{jx}^{(n)} = -\eta\, \varepsilon_j^{(n)} x_j^{(n)} \tag{11}$$

Hidden – Bias:

$$\Delta v_{j0}^{(n)} = -\eta\, \varepsilon_j^{(n)} \tag{12}$$

B. Elman - Recurrent Neural Networks (RNN)

In the RNN architecture, the Backpropagation Through Time (BPTT) Algorithm is used, by adjusting the Elman RNN architecture. [29], [30], [31], [32], [33].

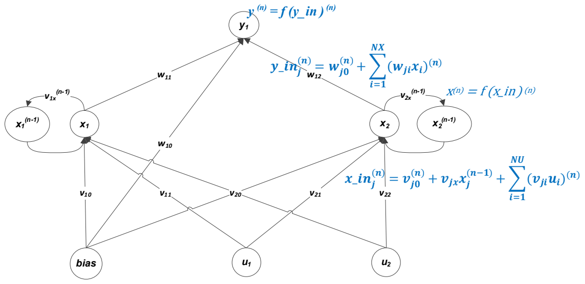

Fig. 3 Feedforward steps of Elman RNN architecture

Feedforward steps of Elman RNN using the BPTT Algorithm as shown in Fig. 3 as follows:

1. Calculate all incoming signals in the *hidden neurons* using equation (13).

$$x\_in_j^{(n)} = v_{j0}^{(n)} + v_{jx} x_j^{(n-1)} + \sum_{i=1}^{NU} (v_{ji} u_i)^{(n)} \tag{13}$$

2. Calculate the output signals of the *hidden neurons* using equation (14).

$$x(n) = f(x\_in)(n) \tag{14}$$

3. Count all incoming signals to the output neurons using equation (15).

$$y\_in_j^{(n)} = w_{j0}^{(n)} + \sum_{i=1}^{NX} (w_{ji} x_i)^{(n)} \tag{15}$$

4. Calculate the output signals of the output neurons using equation (16).

$$y(n) = f(y\_in)(n) \tag{16}$$

5. Calculate the Root Mean Squared Error (RMSE) using equation (17).

$$RMSE = \sqrt{\frac{1}{n}\sum (F_t - A_t)^2} \tag{17}$$

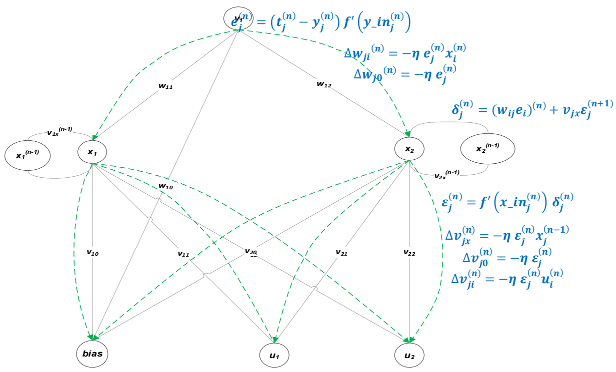

Fig. 4 Backward steps of Elman RNN architecture

Then do the backward steps as shown in Fig. 4 as follows:

6. Calculate the error in the *output units* using equation (18).

$$e_j^{(n)} = \left(t_j^{(n)} - y_j^{(n)}\right) f'\left(y\_in_j^{(n)}\right) \tag{18}$$

7. Calculate weight correction using equation (19) and (20).
Output – Hidden:

$$\Delta w_{ji}{}^{(n)} = -\eta\ e_j^{(n)} x_i^{(n)} \tag{19}$$

Output – Bias:

$$\Delta w_{j0}{}^{(n)} = -\eta\ e_j^{(n)} \tag{20}$$

8. Calculate the sum of the input deltas in the *hidden units* using equation (21).

$$\delta_j^{(n)} = (w_{ij} e_i)^{(n)} + v_{jx} \varepsilon_j^{(n+1)} \tag{21}$$

9. Calculate the error in *hidden units* using equation (22).

$$\varepsilon_j^{(n)} = f'\left(x\_in_j^{(n)}\right) \delta_j^{(n)} \tag{22}$$

10. Calculate weight correction using equations (23), (24), and (25).
Hidden – Input:

$$\Delta v_{jx}^{(n)} = -\eta\ \varepsilon_j^{(n)} x_j^{(n-1)} \tag{23}$$

Hidden – Bias:

$$\Delta v_{j0}^{(n)} = -\eta\ \varepsilon_j^{(n)} \tag{24}$$

Hidden – Hidden (feedback):

$$\Delta v_{jx}^{(n)} = -\eta\ \varepsilon_j^{(n)} x_j^{(n-1)} \tag{25}$$

C. *Jordan* - Recurrent Neural Networks (RNN)

In the RNN architecture, the Backpropagation Through Time (BPTT) Algorithm is used by adjusting the Jordan RNN architecture. [29] [31] [34] [35].

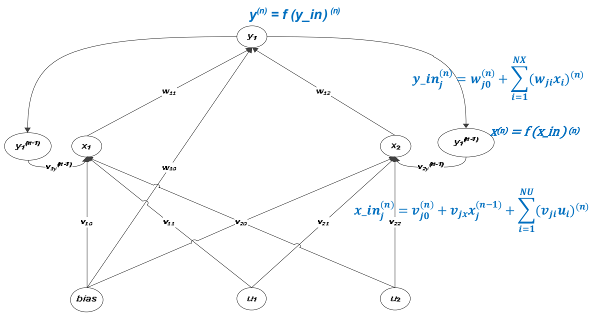

Fig. 5 Feedforward steps of Jordan RNN architecture

Feedforward steps of Jordan RNN using BPTT Algorithm as shown in Fig. 5 as follows:

1. Calculate all incoming signals in the *hidden neurons* using equation (26).

$$x_{in_j}{}^{(n)} = v_{j0}^{(n)} + v_{jx} x_j^{(n-1)} + \Sigma_{i=1}^{NU} (v_{ji} u_i)^{(n)} \tag{26}$$

2. Calculate the output signals of the *hidden neurons* using equation (27).

$$x^{(n)} = f\,(x\_in) \tag{27}$$

3. Count all incoming signals to the *output neurons* using equation (28)

$$y\_in_j^{(n)} = w_{j0}^{(n)} + \Sigma_{i=1}^{NX} (w_{ji} x_i)^{(n)} \tag{28}$$

4. Calculate the output signals of the *output neurons* using equation (29).

$$y^{(n)} = f\,(y\_in) \tag{29}$$

5. Calculate the *Root Mean Squared Error (RMSE)* using equation (30).

$$RMSE = \sqrt{\frac{1}{n} \Sigma (F_t - A_t)^2} \tag{30}$$

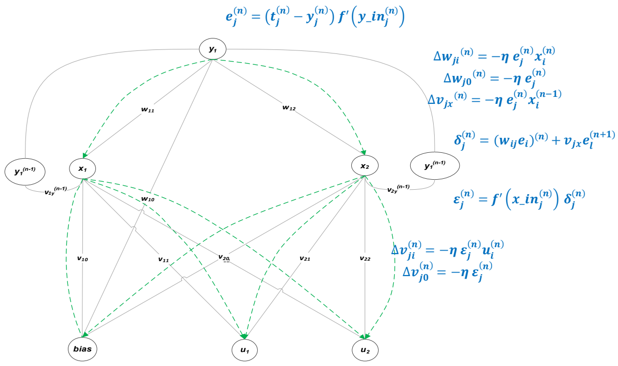

Fig. 6 Backward steps of Elman RNN architecture

Then do the backward steps as shown in Fig. 6 as follows:

6. Calculate the error in the *output units* using equation (31).

$$e_j^{(n)} = \left(t_j^{(n)} - y_j^{(n)}\right) f'\left(y\_in_j^{(n)}\right) \tag{31}$$

7. Calculate weight correction using equations (32), (33), and (34).
Output – Hidden:

$$\Delta w_{ji}{}^{(n)} = -\eta\ e_j^{(n)} x_i^{(n)} \tag{32}$$

Output – Bias:

$$\Delta w_{j0}{}^{(n)} = -\eta\ e_j^{(n)} \tag{33}$$

Output – Hidden (feedback):

$$\Delta v_{jx}{}^{(n)} = -\eta\ e_j^{(n)} x_i^{(n-1)} \tag{34}$$

8. Calculate the sum of the input deltas in the *hidden units* using equation (35).

$$\delta_j^{(n)} = (w_{ij} e_i)^{(n)} + v_{jx} \varepsilon_j^{(n+1)} \tag{35}$$

9. Calculate the error in *hidden units* using equation (36).

$$\varepsilon_j^{(n)} = f'\left(x\_in_j^{(n)}\right) \delta_j^{(n)} \tag{36}$$

11. Calculate weight correction using equations (37) and (38).
Hidden – Input:

$$\Delta v_{ji}^{(n)} = -\eta\ \varepsilon_j^{(n)} u_i^{(n)} \tag{37}$$

Hidden – Bias:

$$\Delta v_{j0}^{(n)} = -\eta\ \varepsilon_j^{(n)} \tag{38}$$

III. Results and Discussion

This system was built using the Java programming language, with the concept of object-oriented programming. The software (Integrated Development Environment) used is Eclipse.

A. Data

In this study, the data used are time series data on currency values and fundamental data that occurs in Indonesia. Time series data, namely the value of the Indonesian Rupiah (IDR) against the US Dollar (USD), is expected to be able to project currency forecasting technical analysis. In comparison, the fundamental analysis used the value of inflation, the money supply (in billions of Rupiah), and the value of exports and imports. Inflation value is the average value of inflation that occurs in all cities in Indonesia. The money supply is the amount of currency, demand deposits, quasi-money, and securities other than shares. Export-Import Value is the export or import value, which is the sum of the oil and gas and non-oil and gas sectors in millions of USD. All fundamental data is obtained from the official website of the Central Statistics Agency https://www.bps.go.id/. Meanwhile, the IDR currency value against USD is taken from the official page of the Bank Indonesia website https://www.bi.go.id/. The data used in this study is data from January 1, 2021, to March 31, 2021 [36], [37], [38], [39], [40], [41].

1) Input Data: This research's file type of input data is Text Document (.txt). The following input data matrix format used in this study is illustrated in Table 1.

TABLE I
INPUT DATA

No	Date	USD Sell	Inflation Rate	Money Supply	Import	Export
1	01/01/2021	14175,53	0,16	6767407,65	13329,9	15293,7
2	01/02/2021	14175,53	0,16	6767407,65	13329,9	15293,7
…	…	…	…	…	…	…
304	10/31/2021	14270,00	0,28	7491704,38	16293,6	22029,7

2) Normalization Data: Before entering the learning process, the input data is processed by normalization so that the value is in the range of 0 to 1. The data resulting from the normalization of the input data is depicted in Fig. 7.

No	Date	USDSell	Inflation	AmountOfMoneyInCirculation(Billion...	Import(Million$)	Export(Million$)
1	01/01/2021	14175.53	0.42	6767407.65	15293.7	13329.9
2	01/02/2021	14175.53	0.42	6767407.65	15293.7	13329.9
3	01/03/2021	14175.53	0.42	6767407.65	15293.7	13329.9
4	01/04/2021	13972.52	0.42	6767407.65	15293.7	13329.9
5	01/05/2021	14014.73	0.42	6767407.65	15293.7	13329.9
6	01/06/2021	13995.63	0.42	6767407.65	15293.7	13329.9
7	01/07/2021	14007.69	0.42	6767407.65	15293.7	13329.9
8	01/08/2021	14128.29	0.42	6767407.65	15293.7	13329.9
9	01/09/2021	14128.29	0.42	6767407.65	15293.7	13329.9
10	01/10/2021	14128.29	0.42	6767407.65	15293.7	13329.9
11	01/11/2021	14225.78	0.42	6767407.65	15293.7	13329.9

Fig. 7 Input data normalization results

3) Training Data: After the normalization process is carried out, the input data is adjusted to the number of input layers. In this study, the number of days in a week is seven days, so the historical data used is D-1 to D-7 for each target. In the system test, 2 test scenarios were carried out, the first using time series data, as shown in Fig. 8, and the second scenario using time series and fundamental data, as shown in Fig. 9.

No	Date	USDSell(t-7)	USDSell(t-6)	USDSell(t-5)	USDSell(t-4)	USDSell(t-3)	USDSell(t-2)	USDSell(t-1)	Inflation)	AmountOfMoneyl...	Import(Million$))	Export(Million$))	USDSell(t)
1	01/09/2021	0.32998800822...	0.32998800822...	0.04017245317...	0.10043113293...	0.07316411603...	0.09038088168...	0.26254853814...	1.0	0.0	0.01210380220...	0.01842441447...	0.26254853814...
2	01/10/2021	0.32998800822...	0.04017245317...	0.10043113293...	0.07316411603...	0.09038088168...	0.26254853814...	0.26254853814...	1.0	0.0	0.01210380220...	0.01842441447...	0.26254853814...
3	01/11/2021	0.04017245317...	0.10043113293...	0.07316411603...	0.09038088168...	0.26254853814...	0.26254853814...	0.26254853814...	1.0	0.0	0.01210380220...	0.01842441447...	0.40172453174...
4	01/12/2021	0.10043113293...	0.07316411603...	0.09038088168...	0.26254853814...	0.26254853814...	0.26254853814...	0.40172453174...	1.0	0.0	0.01210380220...	0.01842441447...	0.51076404751...
5	1/13/2021	0.07316411603...	0.09038088168...	0.26254853814...	0.26254853814...	0.26254853814...	0.40172453174...	0.51076404751...	1.0	0.0	0.01210380220...	0.01842441447...	0.33572693010...
6	1/14/2021	0.09038088168...	0.26254853814...	0.26254853814...	0.26254853814...	0.40172453174...	0.51076404751...	0.33572693010...	1.0	0.0	0.01210380220...	0.01842441447...	0.35007423481...
7	1/15/2021	0.26254853814...	0.26254853814...	0.26254853814...	0.40172453174...	0.51076404751...	0.33572693010...	0.35007423481...	1.0	0.0	0.01210380220...	0.01842441447...	0.27689584285...
8	1/16/2021	0.26254853814...	0.26254853814...	0.40172453174...	0.51076404751...	0.33572693010...	0.35007423481...	0.27689584285...	1.0	0.0	0.01210380220...	0.01842441447...	0.27689584285...
9	1/17/2021	0.26254853814...	0.40172453174...	0.51076404751...	0.33572693010...	0.35007423481...	0.27689584285...	0.27689584285...	1.0	0.0	0.01210380220...	0.01842441447...	0.27689584285...
10	1/18/2021	0.40172453174...	0.51076404751...	0.33572693010...	0.35007423481...	0.27689584285...	0.27689584285...	0.27689584285...	1.0	0.0	0.01210380220...	0.01842441447...	0.29411260849...

Fig. 8 Time series and fundamental training data

No	Date	USDSell(t-7)	USDSell(t-6)	USDSell(t-5)	USDSell(t-4)	USDSell(t-3)	USDSell(t-2)	USDSell(t-1)	USDSell(t)
1	01/09/2021	0.3299880082229343	0.3299880082229343	0.040172453174967426	0.10043113293741467	0.07316411603471891	0.09038088168113463	0.2625485381452737	0.2625485381452737
2	01/10/2021	0.3299880082229343	0.040172453174967426	0.10043113293741467	0.07316411603471891	0.09038088168113463	0.2625485381452737	0.2625485381452737	0.2625485381452737
3	01/11/2021	0.040172453174967426	0.10043113293741467	0.07316411603471891	0.09038088168113463	0.2625485381452737	0.2625485381452737	0.2625485381452737	0.40172453174965866
4	01/12/2021	0.10043113293741467	0.07316411603471891	0.09038088168113463	0.2625485381452737	0.2625485381452737	0.2625485381452737	0.40172453174965866	0.5107640475102786
5	1/13/2021	0.07316411603471891	0.09038088168113463	0.2625485381452737	0.2625485381452737	0.2625485381452737	0.40172453174965866	0.5107640475102786	0.33572693010507026
6	1/14/2021	0.09038088168113463	0.2625485381452737	0.2625485381452737	0.2625485381452737	0.40172453174965866	0.5107640475102786	0.33572693010507026	0.35007423481041666
7	1/15/2021	0.2625485381452737	0.2625485381452737	0.2625485381452737	0.40172453174965866	0.5107640475102786	0.33572693010507026	0.35007423481041666	0.2768958428506175
8	1/16/2021	0.2625485381452737	0.2625485381452737	0.40172453174965866	0.5107640475102786	0.33572693010507026	0.35007423481041666	0.2768958428506175	0.2768958428506175
9	1/17/2021	0.2625485381452737	0.40172453174965866	0.5107640475102786	0.33572693010507026	0.35007423481041666	0.2768958428506175	0.2768958428506175	0.2768958428506175
10	1/18/2021	0.40172453174965866	0.5107640475102786	0.33572693010507026	0.35007423481041666	0.2768958428506175	0.2768958428506175	0.2768958428506175	0.29411260849703064

Fig. 9 Time series training data

B. *Evaluation* of Algorithm Performance in the Training Process

In the training and forecasting process, parameters are used: the number of input layers = 11; the number of hidden layers = 7; the number of output layers = 1; target error = 0.001; learning rate = 0.01; maximum iteration = 4000. An input layer, consisting of the price of IDR against USD during the previous 1 week and the value of the fundamental factor. So the input layer is the price of IDR against USD at (t-7), (t-6), up to (t-1), and the inflation rate, money supply, export value, and import value, which have been normalized. The maximum iteration used is 4000, the activation function used in the hidden layer is the Log-Sigmoid Transfer Function (Logsig), and the activation function used in the output layer is the Linear Transfer Function (Linear). Fig. 10, 11, and 12 are graphs of changes in the results of Error calculations in the training process, using the Backpropagation Algorithm, Elman RNN, and Jordan RNN, using time series and fundamental data.

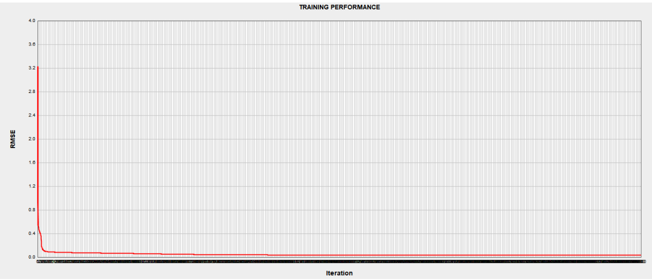

Fig. 10 Graph of RMSE Change in Backpropagation Method Training Process using Time Series and Fundamental Data

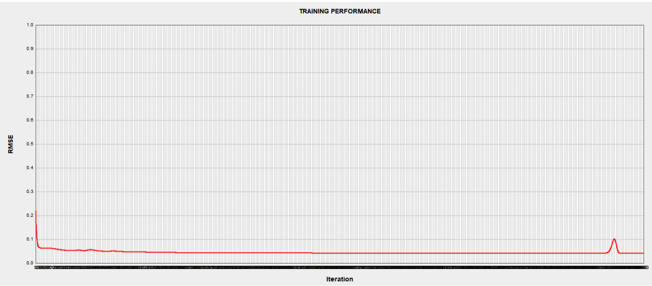

Fig. 11 Graph of RMSE Change in the Elman RNN Method Training Process using Time Series and Fundamental Data

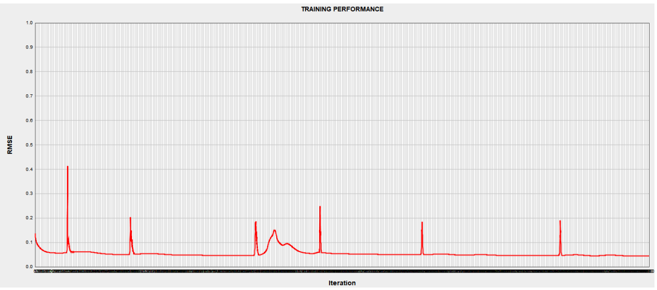

Fig. 12 Graph of Changes in RMSE Jordan RNN Method Training Process using Time Series and Fundamental Data

While Fig. 13, 14, and 15 are graphs of changes in the results of Error calculations in the training process, using the Backpropagation Algorithm, Elman RNN, and Jordan RNN, using time series data.

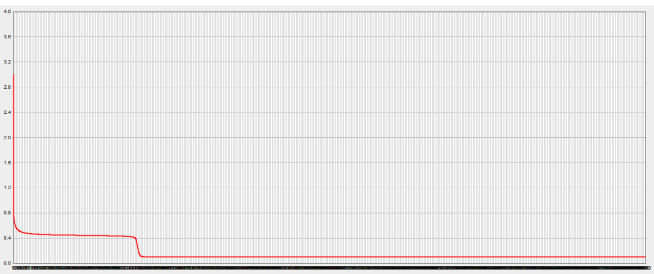

Fig. 13 Graph of RMSE Change in Backpropagation Method Training Process using time series data

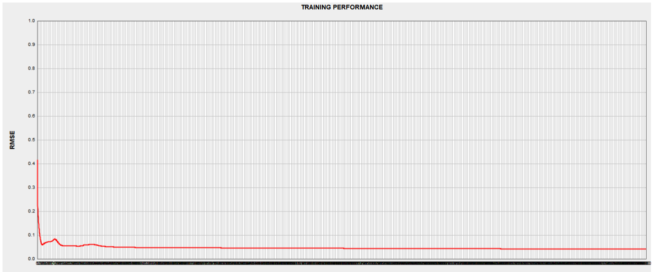

Fig. 14 Graph of RMSE Change in the Elman RNN method of training using time series data

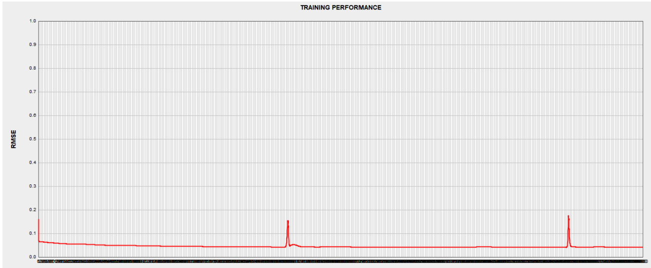

Fig. 15 Graph of Changes in RMSE Jordan RNN Method Training Process using time series data

Fig. 16, 17, and 18 below is a comparison chart of changes in the selling USD value in the training and forecasting process, with the actual USD sell, and the USD sell forecast for the next 7 days. The training and forecasting process uses the Backpropagation Algorithm, Elman RNN, and Jordan RNN, and uses time series and fundamental data. On the chart, the blue line represents the change in the actual selling USD value, the red line represents the change in the selling USD value resulting from the training output, and the black line represents the forecasted selling USD value.

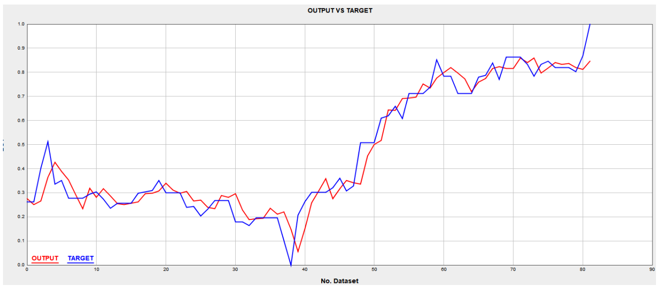

Fig. 16 Comparison Graph of Output Values, Targets, and Forecasting Values, in the Backpropagation Method Training Process, using Time Series and Fundamental Data

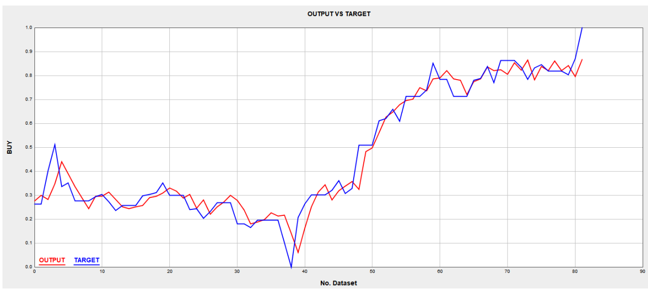

Fig. 17 Comparison Graph of Output Value, Target, and Forecasting Value, in the Elman RNN Method Training Process, using Time Series and Fundamental Data

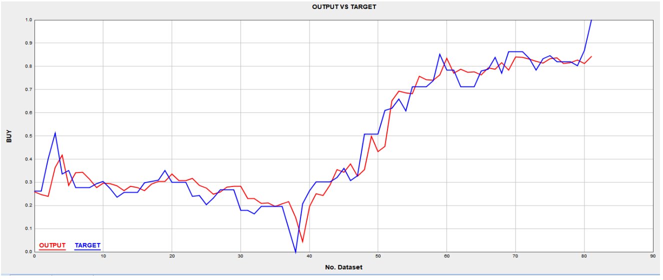

Fig. 18 Graph of Comparison of Output Values, Targets, and Forecasting Values, in the Jordan RNN Method Training Process, using Time Series and Fundamental Data

While Fig. 19, 20, and 21 below is a comparison chart using time series data.

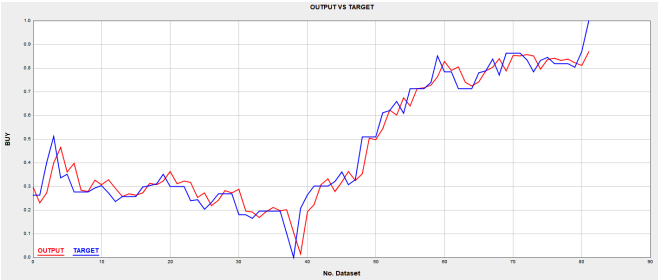

Fig. 19 Comparison Graph of Output Values, Targets, and Forecasting Values, in the Backpropagation Method Training Process, using Time Series Data

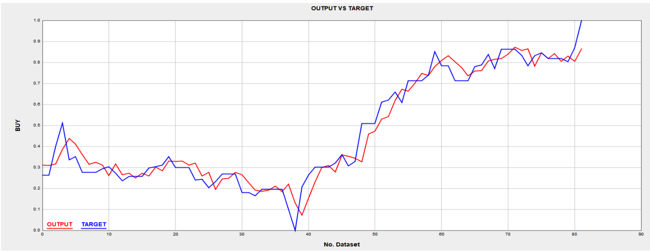

Fig. 20 Comparison Graph of Output Value, Target, and Forecasting Value, in the Elman RNN Method Training Process, using Time Series Data

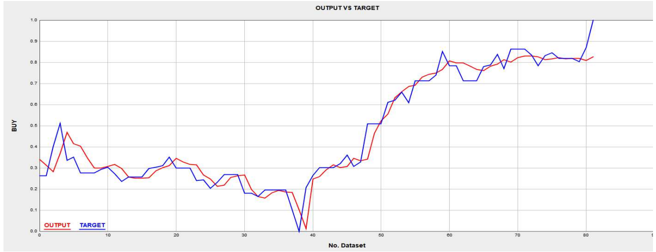

Fig. 21 Graph of Comparison of Output Values, Targets, and Forecasting Values, in the Jordan RNN Method Training Process, using Time Series Data

From the training process the following is an RMSE table from the training process using the Backpropagation, Elman RNN, and Jordan RNN algorithms, with Time Series and Fundamental input data and only Time Series input data.

TABLE II
RMSE RESULT

Data	RMSE		
	Backpropagation	**Elman RNN**	**Jordan RNN**
Time Series + Fundamental	0.042396	0.040596	0.044235
Time Series	0.041185	0.041031	0.04224

From Table II, when using Time series and Fundamental data, it can be concluded that the lowest RMSE is 0.040596 using the Elman RNN algorithm. Meanwhile, when using Time series data, it can be concluded that the lowest RMSE is 0.041031 using the Elman RNN algorithm. When compared based on input data, the RMSE results using Time Series and Fundamental data are lower than using Time Series data.

C. *Evaluation of Algorithm Performance in the Forecasting Process*

In this chapter, the actual value of USD Sell is compared with the forecasting results for the next 7 (seven) days, namely April 1 to 7, 2021, using the Backpropagation, Elman RNN, and Jordan RNN algorithms. Table III results from RMSE using Time Series and Fundamental Data as training data.

TABLE III
RMSE FORECASTING PROCESS USING TIME SERIES AND FUNDAMENTAL DATA

No	Date	USD Sell	Forecasting Result			RMSE		
			Backpropagation	**Elman**	**Jordan**	**Backpropagation**	**Elman**	**Jordan**
1	1-4-2021	14.649,89	14.545,65	14.566,57	14.552,64	104,24	83,32	97,25
2	2-4-2021	14.649,89	14.547,22	14.574,35	14.555,66	103,46	79,53	95,75
3	3-4-2021	14.649,89	14.548,74	14.580,16	14.554,30	102,70	76,40	95,70
4	4-4-2021	14.649,89	14.548,82	14.582,20	14.556,26	102,29	74,32	95,18
5	5-4-2021	14.656,92	14.546,82	14.584,56	14.549,90	103,90	73,93	97,67
6	6-4-2021	14.605,67	14.546,49	14.587,21	14.552,92	97,88	67,91	91,72
7	7-4-2021	14.591,60	14.547,28	14.591,30	14.553,71	92,15	62,87	86,11

TABLE IV
RMSE FORECASTING PROCESS USING TIME SERIES DATA

No	Date	USD Sell	Forecasting Result			RMSE		
			Backpropagation	**Elman**	**Jordan**	**Backpropagation**	**Elman**	**Jordan**
1	1-4-2021	14.649,89	14.487,16	14.497,57	14.487,13	162,73	152,32	162,76
2	2-4-2021	14.649,89	14.439,03	14.477,90	14.472,23	149,10	162,45	170,37
3	3-4-2021	14.649,89	14.405,40	14.469,12	14.458,41	188,34	168,78	177,69
4	4-4-2021	14.649,89	14.372,82	14.467,83	14.448,72	208,74	172,20	183,84
5	5-4-2021	14.656,92	14.339,32	14.460,14	14.434,53	227,76	177,38	192,17
6	6-4-2021	14.605,67	14.316,63	14.462,83	14.440,09	248,34	172,11	188,00
7	7-4-2021	14.591,60	14.293,04	14.461,70	14.435,66	255,57	166,74	183,76

Table III shows that the lowest RMSE uses the Elman RNN algorithm in the forecasting process using Time Series and Fundamental data as training data. The table is the result of RMSE using Time Series Data as training data. Table IV shows that the lowest RMSE is using the Elman RNN algorithm in the forecasting process using only Time Series data as training data. From the two RMSE tables above, the lowest RMSE in the forecasting process is using time series and fundamental data as training data, compared to using only Time Series data, using the Elman RNN algorithm. Although the fundamental value data provided by the Indonesian government is not too volatile or updated every month, the USD time series data is updated daily.

IV. Conclusion

In this study, the problem raised is forecasting the Indonesian Rupiah (IDR) against the US dollar (USD) using input data, namely time series data and fundamental data in Indonesia. Time series and fundamental data used as training data are from January 1 to March 31, 2021, while the forecasting data to be compared with actual data is from April 1 to 7, 2021. The algorithms used and compared in this study are Backpropagation, Elman Recurrent Neural Network (Elman RNN), and Jordan Recurrent Neural Network (Jordan RNN). Root Mean Squared Error (RMSE) is used to measure system performance. After the training process is carried out, the highlights can be listed as follows: The Elman RNN algorithm produces the lowest RMSE in the training process using Time Series and Fundamental Data, compared to Backpropagation and Jordan RNN. The Elman RNN algorithm produces the lowest RMSE value in the training process using only Time Series data, compared to Backpropagation and Jordan RNN. The RMSE of the Elman RNN Algorithm using Time Series and Fundamental Data is lower than the RMSE of the Elman RNN Algorithm in the training process using Time Series Data only. The Elman RNN algorithm produces the lowest RMSE in the forecasting process using time series and fundamental data, compared to Backpropagation, and Jordan RNN. The Elman RNN algorithm produces the lowest RMSE in the forecasting process using only Time Series data, compared to Backpropagation, and Jordan RNN. The RMSE results of the Elman RNN Algorithm using time series and fundamental data are lower than the RMSE results of the Elman RNN Algorithm in the forecasting process using time series data only. So, it can be concluded that the performance of the Elman RNN Algorithm is better than the Backpropagation, and Jordan RNN in the case of forecasting the IDR currency against USD using time series and fundamental data.

References

[1] F. Leonard and H. Akbar, "Coffee Granularity Classification using Convolutional Neural Network (CNN)," *Journal of Applied Science, Engineering, Technology, and Education,* vol. IV, no. 1, pp. 133-145, 2022. doi: 10.35877/454RI.asci842.

[2] Y. You and X. Liu, "Forecasting short-run exchange rate volatility with monetary fundamentals: A GARCH-MIDAS approach," *Journal of Banking & Finance,* 2020. doi: 10.1016/j.jbankfin.2020.105849.

[3] Y. Tan, Z. Wang, H. Xiong and Y. Liu, "Fundamental momentum and enhanced fundamental momentum: Evidence from the Chinese stock market," *International Review of Economics & Finance,* pp. 680-693, 2022. doi: 10.1016/j.iref.2022.02.012.

[4] M. D. Bianco, M. Camacho and G. P. Quiros, "Short-run forecasting of the euro-dollar exchange rate with economic fundamentals," *Journal of International Money and Finance,* pp. 377-396, 2012. doi: 10.1016/j.jimonfin.2011.11.018.

[5] M. M. Kumbure, C. Lohrmann, P. Luukka and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," *Expert Systems With Applications,* 2022. doi: 10.1016/j.eswa.2022.116659.

[6] O. F. Beyca, B. C. Ervural, E. Tatoglu, P. G. Ozuyar and S. Zaim, "Using machine learning tools for forecasting natural gas consumption in the province of Istanbul," *Energy Economics,* 2019. doi: 10.1016/j.eneco.2019.03.006.

[7] L. Nevasalmi, "Forecasting multinomial stock returns using machine learning methods," *The Journal of Finance and Data Science,* pp. 86-106, 2020. doi: 10.1016/j.jfds.2020.09.001.

[8] A. Dubois, F. Teytaud and S. Verel, "Short term soil moisture forecasts for potato crop farming: A machine learning approach," *Computers and Electronics in Agriculture,* 2021. doi: 10.1016/j.compag.2020.105902.

[9] G. K. Vishwakarma, C. Paul and A. M. Elsawah, "An algorithm for outlier detection in a time series model using backpropagation neural network," *Journal of King Saud University,* pp. 3328-3336, 2020. doi: 10.1016/j.jksus.2020.09.018.

[10] M. Elbayoumi, N. A. Ramli, N. Faizah and F. M. Yusof, "Development and comparison of regression models and feedforward backpropagation neural network models to predict seasonal indoor PM2.5–10 and PM2.5 concentrations in naturally ventilated schools," *Atmospheric Pollution Research,* pp. 1013-1023, 2015. doi: 10.1016/j.apr.2015.09.001.

[11] R. Olawoyin, "Application of backpropagation artificial neural network prediction model for the PAH bioremediation of polluted soil," *Chemosphere,* pp. 145-150, 2016. doi: 10.1016/j.chemosphere.2016.07.003.

[12] C. Liu, "Risk Prediction of Digital Transformation of Manufacturing Supply Chain Based on Principal Component Analysis and Backpropagation Artificial Neural Network," *Alexandria Engineering Journal,* vol. 61, no. 1, pp. 775-784, 2022. doi: 10.1016/j.aej.2021.06.010.

[13] Y.-L. Wang, H. Jahanshahi, S. Bekiros, F. Bezzina, Y.-M. Chu and A. A. Alyh, "Deep recurrent neural networks with finite-time terminal sliding mode control for a chaotic fractional-order financial system with market confidence," *Chaos, Solitons and Fractals,* vol. 146, 2021.

[14] Y. Song and Y. Wang, "A big-data-based recurrent neural network method for forest energy estimation," *Sustainable Energy Technologies and Assessments,* vol. 55, 2023. doi: 10.1016/j.seta.2022.102910.

[15] M. Villegas, A. Gonzalez-Agirre, A. Gutiérrez-Fandiño, J. Armengol-Estapé, C. P. Carrino, D. Pérez-Fernández, F. Soares, P. Serrano, M. Pedrera, N. García and A. Valencia, "Predicting the evolution of COVID-19 mortality risk: A Recurrent Neural Network approach," *Computer Methods and Programs in Biomedicine,* vol. 3, 2023. doi: 10.1016/j.cmpbup.2022.100089.

[16] P. R. Vlachas, J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott and P. Koumoutsakosa, "Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics," *Neural Networks,* vol. 126, pp. 191-217, 2020. doi: 10.1016/j.neunet.2020.02.016.

[17] M. S. Alhajeri , Z. Wu, D. Rincon, F. Albalawi and P. D. Christofides, "Estimation-Based Predictive Control of Non-linear Processes Using Recurrent Neural Networks," *IFAC PapersOnLine,* vol. 54, no. 3, p. 91–96, 2021. doi: 10.1016/j.ifacol.2021.08.224.

[18] X. Zhang, C. Zhong, J. Zhang, T. Wang and W. W. Y. Ng, "Robust Recurrent Neural Networks for Time Series Forecasting," *Neurocomputing,* 2023. doi: 10.1016/j.neucom.2023.01.037.

[19] S. Li and Y. Yang, "A recurrent neural network framework with an adaptive training strategy for long-time predictive modeling of non-linear dynamical systems," *Journal of Sound and Vibration,* vol. 506, 2021. doi: 10.1016/j.jsv.2021.116167.

[20] H. Hewamalage, C. Bergmeir and K. Bandara, "Recurrent Neural Networks for Time Series Forecasting: Current status and future directions," *International Journal of Forecasting,* vol. 37, no. 1, pp. 388-427, 2021. doi: 10.48550/arXiv.1909.00590.

[21] Y. Wang, L. Wang, F. Yang, W. Di and Q. Chang, "Advantages of direct input-to-output connections in neural networks: The Elman network for stock index forecasting," *Information Sciences,* pp. 1066-1079, 2021. doi: 10.1016/j.ins.2020.09.031.

[22] Q. Meng, X. Yuan, X. Ren, H. Li, L. Jiang and L. Yang, "Thermal Energy Storage Air-conditioning Demand Response Control Using Elman Neural Network Prediction Model," *Sustainable Cities and Society,* 2022. doi: 10.1016/j.scs.2021.103480.
[23] A. Zaras, N. Passalis and A. Tefas, Deep Learning for Robot Perception and Cognition, Academic Press, 2022.
[24] L. S. Moonlight and A. S. Prabowo, "Forecasting System for Passenger, Airplane, Luggage and Cargo, Using Artificial Intelligence Method-Backpropagation Neural Network at Juanda International Airport," *Warta Ardhia Jurnal Perhubungan Udara,* vol. 45, no. 2, pp. 99-110, 2019. doi: 10.25104/wa.v45i2.358.99-110.
[25] L. S. Moonlight, F. Faizah, Y. Suprapto and N. Pambudiyatno, "Comparison of Backpropagation and Kohonen Self Organising Map (KSOM) Methods in Face Image Recognition," *Journal of Information Systems Engineering and Business Intelligence,* vol. 7, no. 2, pp. 149-161, 2021. doi: 10.20473/jisebi.7.2.149-161.
[26] A. Y. Alanis, N. Arana-Daniel and C. López-Franco, Artificial Neural Networks for Engineering Applications, Guadalajara, Mexico: Academic Press, 2019.
[27] A. Subasi, Practical Machine Learning for Data Analysis Using Python, Jeddah, Saudi Arabia: Academic Press, 2020.
[28] M. Gori, Machine Learning - A Constraint-Based Approach, Morgan Kaufmann, 2018.
[29] K. Bandara, C. Bergmeir and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Systems with Applications,* vol. 140, 2020. doi: 10.48550/arXiv.1710.03222.
[30] S. Achanta and S. V. Gangashetty, "Deep Elman recurrent neural networks for statistical parametric speech synthesis," *Speech Communication,* vol. 93, pp. 31-42, 2017. doi: 10.1016/j.specom.2017.08.003.
[31] A. Glassner, Deep Learning : A Visual Approach, San Francisco: Willian Pollock, 2021.
[32] J. D. Rios, A. Y. Alanis, N. Arana-Daniel, C. Lopez-Franco and E. N. Sanchez, Neural Networks Modeling and Control - Applications for Unknown Non-linear Delayed Systems in Discrete Time, Jalisco, Mexico: Academic Press, 2020.
[33] A. Mechelli and S. Vieira, Machine Learning - Methods and Applications to Brain Disorders, London, United Kingdom: Academic Press, 2019.
[34] Z. Al-Ghazawi and R. Alawneh, "Use of artificial neural network for predicting effluent quality parameters and enabling wastewater reuse for climate change resilience – A case from Jordan," *Journal of Water Process Engineering,* vol. 44, 2021. doi: 10.1016/j.jwpe.2021.102423.
[35] P. Kumar, Y. Kumar and M. A. Tawhid, Machine Learning, Big Data, and IoT for Medical Informatics, Academic Press, 2021.
[36] R. L. Kissell, Algorithmic Trading Methods - Applications Using Advanced Statistics, Optimization, and Machine Learning Techniques, NY, United States: Academic Press, 2020.
[37] BPS, "Indeks Harga Konsumen dan Inflasi Bulanan Indonesia," 20 January 2022. [Online]. Available: https://www.bps.go.id/statictable/2009/06/15/907/indeks-harga-konsumen-dan-inflasi-bulanan-indonesia-2006-2022.html.
[38] BPS, "Nilai Ekspor Migas-NonMigas," 15 January 2022. [Online]. Available: https://www.bps.go.id/indicator/8/1753/1/nilai-ekspor-migas-nonmigas.html.
[39] BPS, "Nilai Impor Migas-NonMigas," 10 January 2022. [Online]. Available: https://www.bps.go.id/indicator/8/1754/2/nilai-impor-migas-nonmigas.html.
[40] BI, "Bank Indonesia - Bank Sentral Republik Indonesia," 5 January 2022. [Online]. Available: https://www.bi.go.id/id/statistik/informasi-kurs/transaksi-bi/default.aspx.
[41] BPS, "Uang Beredar," 8 January 2022. [Online]. Available: https://www.bps.go.id/indicator/13/123/1/uang-beredar.html.