JAMBURA JOURNAL OF MATHEMATICS

Jambura J. Math. Vol. 5, No. 2, pp. 419–430, August 2023

Journal Homepage: http://ejurnal.ung.ac.id/index.php/jjom

DOI: https://doi.org/10.34312/jjom.v5i2.20407

Implementation of Hybrid RNN-ANFIS on Forecasting Jakarta Islamic Index

Yogi Anggara[1], **Arif Munandar**[1,*]

[1]*Department of Mathematics, Faculty of Mathematics and Natural Sciences,UIN Sunan Kalijaga, Yogyakarta 55281, Indonesia*

[*]*Corresponding author. Email: arif.munandar@uin-suka.ac.id*

ABSTRACT

RNN is a type of artificial neural network used to handle problems that require sequential data processing. ANFIS is a method that combines the advantages of fuzzy logic and artificial neural networks to create a system, so can adapt the parameters it uses according to the obtained data so that it can build an automated inference system. In this research, we make combination of RNN in ANFIS, which makes ANFIS able to accept input in the form of time series data so that ANFIS can recognize patterns contained in the time series data and its suitable for forecasting cases in the Jakarta Islamic Index. The membership functions used are three Gaussian functions. The results of the RNN-ANFIS Hybrid model training provide an interpretation that the first membership function represents the trend change indicator value, the second membership function represents the closing price change value in the last eight days, and the third membership function represents the pattern change value in the trend. The model for the Jakarta Islamic Index provides very good predictions with an MSE value of 0.001 and an MAE of 0.246.

Keywords:

RNN-ANFIS; Neural Network; Fuzzy Logic; Jakarta Islamic Index

Citation Style:

Y. Anggara and A. Munandar, "Implementation of Hybrid RNN-ANFIS on Forecasting Jakarta Islamic Index", *Jambura J. Math.*, vol. 5, No. 2, pp. 419–430, 2023, doi: https://doi.org/10.34312/jjom.v5i2.20407

1. Introduction

In the 20th century, a new concept founded in mathematical logic named as fuzzy logic, which developed by Lotfi A. Zadeh in 1965 [1]. Fuzzy logic is a system of logic that allows level of truth that is not only worth right or wrong, but also value between right and wrong. Unlike Boolean logic where an approach to computing only based on "true of false", fuzzy logic using an approach to computing base on "degrees of truth". Because of this ability, fuzzy logic is able to make the system's capabilities on the sensor to be smoother. For example, an automatic brightness sensor on a device, fuzzy logic is able to determine what percentage of the brightness level on the device based on the level of light intensity, whether the environment is dark, bright, very bright or other types.

In the early 1990s Jang on research [2] and [3] created an artificial neural network based on the Takagi-Sugeno fuzzy inference system [4], named with the Adaptive

e-ISSN: 2656-1344

Received: 2023-06-12 | Revised: 2023-07-31 | Accepted: 2023-08-05| Published: 2023-08-07

Neuro-Fuzzy Inference System (ANFIS). ANFIS has ability to adjust the problems it faces and improve its predictive ability constantly. Since that time, ANFIS become one of the most popular methods for solving uncertain problems by using artificial neural networks. ANFIS has been widely used in various fields, such as control systems, classifications, pattern recognitions or forecasting and other fields. Applications in the field of control systems can be found in [5], while [6] given detail application of control non linear systems. Applications in the field of classifications can be seen in research [7], while [8] make an detail on electromyogram (EMG) signal. Applications for the field of pattern recognitions or forecasting can be found on [9, 10]. ANFIS is constantly evolving and optimized to be one of the best methods for resolving uncertain problems by using artificial neural networks.

Recurrent Neural Network (RNN) is a type of neural network that has ability to remember past information and use it to making decisions in the future. RNN was first introduced by Sepp Hochreiter and Jurgen Schmidhuber in 1997 through his research in [11]. Before the advent of RNN, neural networks could only process incoming information linearly, without regard to past information. This limits the ability of neural networks to understand contextual information, such as language, music, or data patterns that have a relationship with data in the past. To solve this problem, Hochreiter and Schmidhuber developed the concept of a neural network that can remember past information through a unit called a ”hidden state”. This hidden state is a type of memory that can be read and written by neural networks, so that the network can use past information as a basis for processing new information. With the presence of RNN, neural networks begin to be able to understand contextual information better and predict future events more accurately, some research that use RNN can be found in [12–15].

Inspired by several research that use hybrid methods by combining some methods such as [16], [17] or [18] and look the potential of RNN and ANFIS, we see that this can be an opportunity to create a hybrid model between RNN and ANFIS. RNN that is able to process input data in the form of sequential data allows ANFIS to learn the fuzzy set membership function automatically on input data in the form of time-series data. This is very helpful for the prediction process, especially in data where future values are influenced by past data patterns such as stocks. In this research, we already implemented RNN-ANFIS in the Jakarta Islamic Index (JII). JII is an index for Sharia stocks in Indonesia, where the value of JII is able to project the movement of Sharia stock prices as a whole. In this research, we use data from the Jakarta Islamic Idex from January 6, 2020 to December 9, 2022, and the output data that will be predicted is the close price data one day ahead. Hopefully, this research can help investors who want to invest in Sharia stock instruments.

2. Methods

This reseach used a method that combined RNN and ANFIS models. The obtained model is then used to predict stock data from the Jakarta Islamic Index. Details of the data used along with the RNN and ANFIS models are presented as follows:

2.1. Data

The data used in this resach is secondary data derived from the Yahoo Finance website [19]. The data was taken in the range of January 6, 2020 to December 9, 2022. Input data on the model is the first eight lags of open price (X_1={$x_{1,1}$, $x_{1,2}$,. . ., $x_{1,8}$}), height price (X_2={$x_{2,1}$, $x_{2,2}$,. . ., $x_{2,8}$}), low price (X_3={$x_{3,1}$, $x_{3,2}$,. . ., $x_{3,8}$}), and close price (X_4={$x_{4,1}$, $x_{4,2}$,. . ., $x_{4,8}$}).

Table 1. Data of Jakarta Islamic Index January 6, 2020 – December 9, 2022

No.	Date	Open	High	Low	Close
1	January 6, 2020	695.98	696.95	689.07	690.06
2	January 7, 2020	691.17	693.78	686.57	692.54
3	January 8, 2020	688.33	688.95	683.72	685.01
4	January 9, 2020	687.79	691.38	686.54	691.38
⋮	⋮	⋮	⋮	⋮	⋮
715	December 6, 2022	599.75	599.75	585.02	587.89
716	December 7, 2022	587.89	590.54	584.51	590.34
717	December 8, 2022	590.18	592.36	579.08	592.36
718	December 9, 2022	592.36	592.36	581.42	583.93

Table 1 is JII data for January 6 2020 to December 9 2022. For example, based on this table, the first input for an open price is $X_1 = \{695.98,\ 691.17,\ 688.33,\ 687.79,\ \ldots\}$, the first input for a high price is $X_2 = \{696.95,\ 693.78,\ 688.95,\ 691.38,\ \ldots\}$, the first input for a low price is $X_3 = \{689.07,\ 686.57,\ 688.95,\ 691.38,\ \ldots\}$, and the first input for a close price is $X_4 = \{690.06,\ 692.54,\ 685.01,\ 691.38,\ \ldots\}$.

2.2. Recurrent Neural Network (RNN)

Recurrent neural networks or RNN are a family of neural networks used to process sequential data [20]. The way recurrent neural networks work is similar to the convolution layer, the difference is that the convolution layer generally used to process input data sets in the form of images, while recurrent neural networks are used to process value sets such as $\boldsymbol{x}^{(1)}$, ..., $\boldsymbol{x}^{(h)}$ [21]. The symbol $\boldsymbol{x}^{(1)}$, ..., $\boldsymbol{x}^{(h)}$ are element of R^d, where d is the number of features. RNN has a unique characteristic, namely that its architecture has at least one feedbackloop, so it can store data in its network structure, thus the performance of RNN in making predictions depends on its weight and architecture [22]. Another uniqueness of RNN is the presence of a feedback connection that carries noise information from the previous input and can be accommodated for the next input [23]. Mathematically RNN can be written as follows [21]:

$$\boldsymbol{a}^{(t)}=\boldsymbol{b}+\boldsymbol{W}\boldsymbol{h}^{(t-1)}+\boldsymbol{U}\boldsymbol{x}^{(t)}\in\mathbb{R}^h \tag{1}$$

$$\boldsymbol{h}^{(t)}=\tanh\left(\boldsymbol{a}^{(t)}\right)\in(-1,1)^h \tag{2}$$

$$\boldsymbol{o}^{(t)}=\boldsymbol{c}+\boldsymbol{V}\boldsymbol{h}^{(t)}\in\mathbb{R}^h \tag{3}$$

$$\hat{\boldsymbol{y}}^{(t)}=\text{softmax}\left(\boldsymbol{o}^{(t)}\right)\in(0,1)^h \tag{4}$$

The parameters consist of the bias vectors $\boldsymbol{b},\ \boldsymbol{c}\in\mathbb{R}^h$ along with the weight matrices $\boldsymbol{U}\in\mathbb{R}^{h\times d}$, $\boldsymbol{V}$, $\boldsymbol{W}\in\mathbb{R}^{h\times h}$. Each of those equations above represent connection between input-to-hidden, hidden-to-output, and hidden-to-hidden.

Figure 1 is an illustration of how the RNN calculation process occurs, where the output value from the previous period's RNN calculation will be forwarded to the RNN

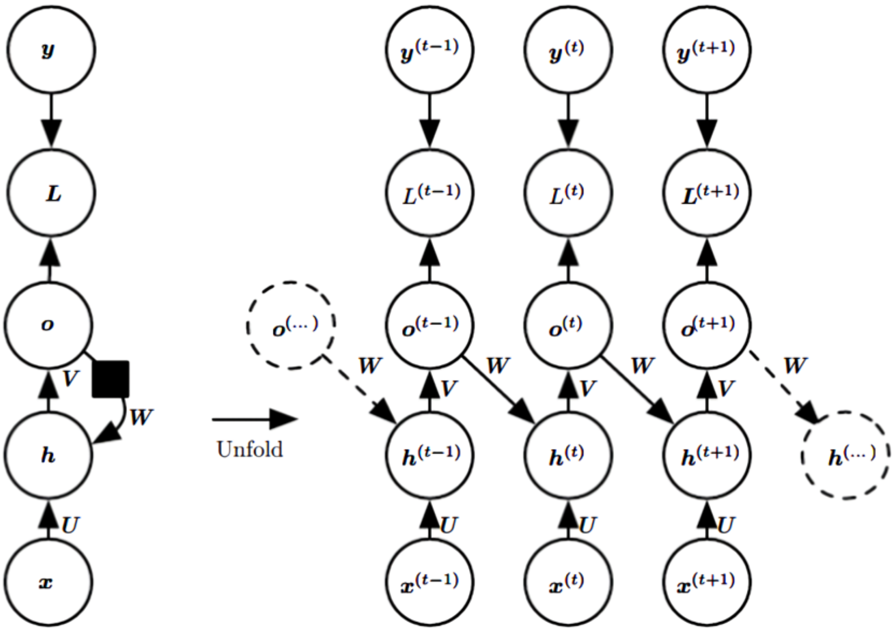

Figure 1. Illustration of the RNN architecture works

calculation in the next period. The above system of equations is an example of a network that continuously maps input sequences to output sequences of the same length. The total loss for a set of $\boldsymbol{x}$ values paired with a set of $\boldsymbol{y}$ values will be the sum of the losses across all time steps. For example, if $L(t)$ is the logarithmic values of the negative probability $\boldsymbol{y}(t)$ with the input data $\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(t)}$, then mathematically the equation is as follows:

$$\begin{aligned} L\left(\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(t)}\},\{\boldsymbol{y}^{(1)},\dots,\boldsymbol{y}^{(t)}\}\right) &= \sum_t L^{(t)} \\ &= -\sum_t \log p_{model}\left(\boldsymbol{y}^{(t)} \;\middle|\; \{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(t)}\}\right). \end{aligned} \tag{5}$$

In this case, the model i.e. $p_{model}\left(\boldsymbol{y}^{(t)} \;\middle|\; \boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(t)}\right)$ is obtained by reading the $\boldsymbol{y}(t)$ entry from the model output vector $\hat{\boldsymbol{y}}(t)$ in equation (4). For a simple form of RNN, the model is condensed into equation (1) only [24].

2.3. Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS is a fuzzy inference system consisting of artificial neuron network algorithms and fuzzy logic that can be used to solve complex nonlinear problems. The ANFIS architecture, namely the adaptive network-based fuzzy inference system, is a fuzzy inference system implemented in the form of an adaptive network. Using hybrid learning methods, ANFIS can build input-output mappings based on fuzzy if-then rules formed by human knowledge and established input-output data pairs [25]. This system can be divided into two groups, first group consisting of neural networks with fuzzy weights and fuzzy activation functions, and the second group consisting of neural networks with inputs formed into fuzzy sets in the first or second layer, but the weights on those neural networks are not formed into fuzzy sets [26].

Functionally ANFIS has the same architecture as the Fuzzy rule base model Sugeno. The architecture of ANFIS is also same as the neural networks with radial functions and certain limitations. ANFIS can be thought as a method that uses learning algorithms against a set of data to adjust rules, so ANFIS also allows those rules to adapt. For examples, 4 input data are given namely x_1, x_2, x_3, x_4, and the fuzzy membership set of each input data is 3. If the membership function of input data uses the Gaussian function, then mathematically ANFIS can be written as follows [27]:

$$\pi_{ij}\left(x_i\right) = N\left(x_i \mid \mu, \sigma\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2\right) \tag{6}$$

$$w_i = \prod_{j=1}^{3} \pi_{ij} \tag{7}$$

$$\overline{w}_i = w_i \left(\sum_{j=1}^{4} w_j\right)^{-1} \tag{8}$$

$$\overline{w}_i f_i = \overline{w}_i \sum_{j=1}^{4} \left(a_j x_j + b_j\right) \tag{9}$$

$$O = \sum_{i=1}^{4} \overline{w}_i f_i \tag{10}$$

Equation (6) is a fuzzification layer where each neuron adapts to the activation parameters. Each neuron generates a membership degree using a Gaussian function as the input membership function. This equation is the first layer in Figure 2. Equation (7) is a layer consisting of fixed neurons which is the product of all input values. Usually the AND operator is used. The result of this calculation is called the firing strength of a rule. Each neuron represents rule i.

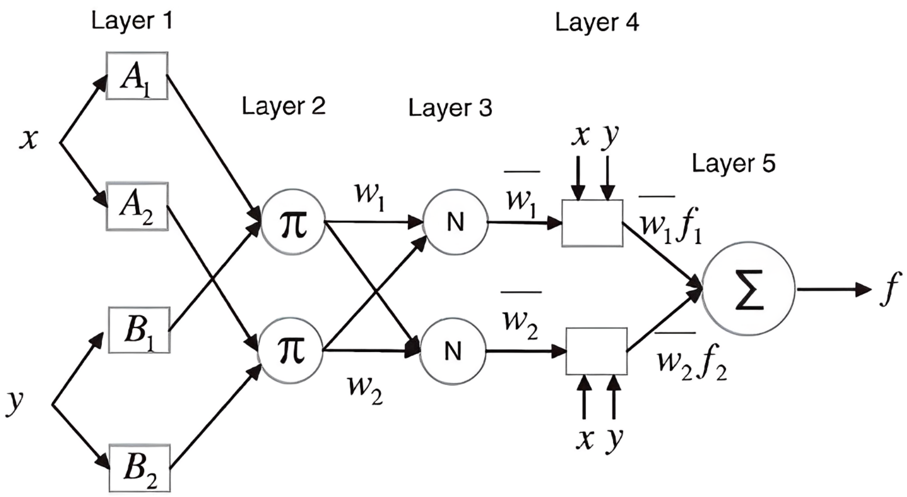

Figure 2. ANFIS architecture

This equation is the second layer in Figure 2. Equation (8) is a layer consisting of neurons which aims to calculate the ratio of the i-th firing strength (w_i) to the total firing strength in the second layer. The result of this calculation is called normalized firing strength. This equation is the third layer in Figure 2. Equation (9) is a layer consisting of neurons that are adaptive to certain outputs. The calculation results for this layer involve the normalized firing strength (w_i) of the third layer, as well as the p_i, q_i, and r_i parameters of the neurons. These parameters are usually called consequent parameters. This equation is the fourth layer in Figure 2. Equation (10) is a layer consisting of a single neuron that produces the sum of all outputs in the fourth layer. In this model, what will be trained are the a_j weight, b_j bias and the parameters of gausian function μ and σ. The output result is the entire summation of equation to equation (9).

3. Results and Discussions

3.1. RNN-ANFIS Architecture

In this research, RNN was used as a layer to study trend patterns of input data in the form of time-series data, namely the first eight lags at open price, height price, low price, and close price.Furthermore, the RNN layer will provide one output value for each input data which is a representation of the pattern on the trend. The output value is then processed on ANFIS to forecast the close price one day ahead. The modeling in this research used the TensorFlow library available on the Python language, while the program code for the ANFIS architecture used code from previous research conducted by Lenhard [28]. We modified the code so that ANFIS can receive output values from RNN and carry out the RNN and ANFIS training process simultaneously. The relationships between the layers are represented on the plot in Figure 3.

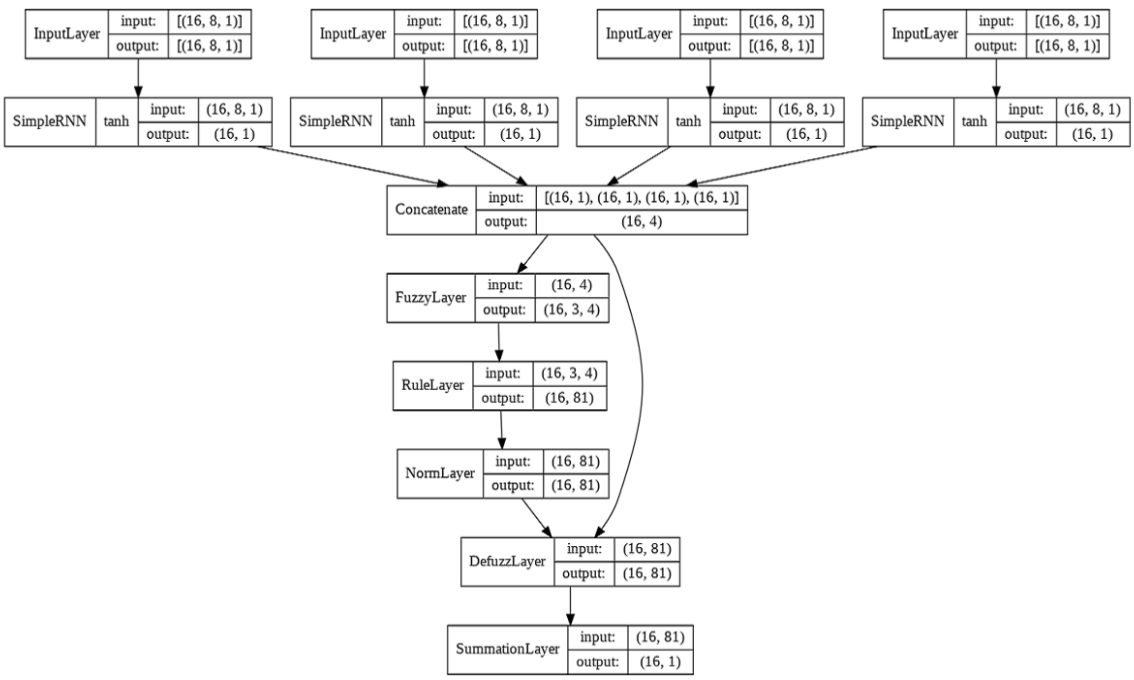

Figure 3. RNN-ANFIS architecture

The steps for RNN-ANFIS model in this research are as follows:

1. Scale each input data using a min-max scaler to have the same range

2. Convert data into the form of supervised learning, which is to have input and output data, then divide the data into two parts, namely 90% for training data and validation data for the training process, and the remaining 10% for test data.
3. Define the RNN-ANFIRIS architecture.
4. Train the model using the Stochastic Gradient Descent Algorithm.
5. Evaluate the model using Means Squared Error (MSE) and Mean Absolute Error (MAE). If the model is still not good, go back to step c.

3.2. Application of RNN-ANFIS on JII

In this reseach, the training process was carried 25 epochs, and each epoch consisted of 25 batches. In Figure 4, although the MSE change value looks not stable, the model is overall able to lower the MSE value. This shows that the RNN-ANFIS training process is good enough.

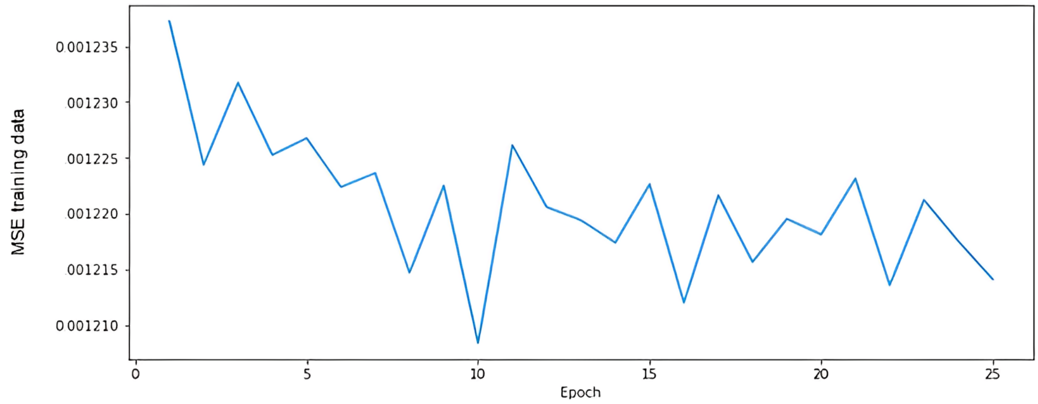

Figure 4. Plot MSE values of RNN-ANFIS training per epoch

In Figure 5, the overall magnitude of the MAE decrease is not as significant as in the MSE decrease. This may be because MAE basically uses the absolute value of error, while MSE uses the square of error. Base on this reason when the error value is less than one, the error value of MSE tends to be smaller. However, the MAE value still decreases with each addition of epoch.

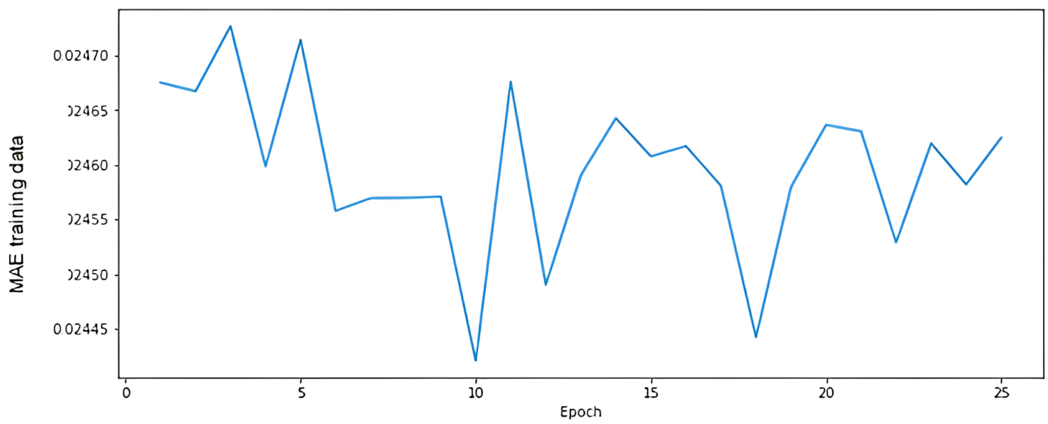

Figure 5. Plot MAE values of RNN-ANFIS training per epoch

The training process that has been carried out produces an RNN-ANFIS model with optimal parameters as follows:

1. **RNN Parameters**
   RNN is the first layer entered by each input data. Each independent variable is entered in one RNN neuron. Thus, there are four neurons in the RNN layer.
   (a) Input data of open price
   $$h_{1,t}= 0,760x_{1,t}+1,061h_{1,t-1}+0,109 \tag{11}$$
   (b) Input data of height price
   $$h_{2,t}= 0,760x_{1,t}+1,061h_{1,t-1}+0,109 \tag{12}$$
   (c) Input data of low price
   $$h_{3,t}= 0,180x_{1,t}+0,941h_{1,t-1}+0,034 \tag{13}$$
   (d) Input data of close price
   $$h_{4,t}= 1,007x_{1,t}+0,969h_{1,t-1}-0,057 \tag{14}$$
2. **Membership functions**
   Each neuron in the RNN is taken for the last hidden state period, which is $t_{i,8}, i = 1,2,3,4$. That is, each input consisting of eight time series data is represented by one value. Next, the hidden state of the last period is forwarded to the fuzification layer.
   (a) Input data of open price
   $$\mu_{1,1}=N\left(h_{1,8} \mid -1,500; 0,843\right) \tag{15}$$
   $$\mu_{1,2}=N\left(h_{1,8} \mid -0,212; 1,191\right) \tag{16}$$
   $$\mu_{1,3}=N\left(h_{1,8} \mid 1,470; 1,184\right) \tag{17}$$
   (b) Input data of height price
   $$\mu_{1,1}=N\left(h_{2,8} \mid -1,488; 1,261\right) \tag{18}$$
   $$\mu_{1,2}=N\left(h_{2,8} \mid -0,003; 0,743\right) \tag{19}$$
   $$\mu_{1,3}=N\left(h_{2,8} \mid 1,501; 1,143\right) \tag{20}$$
   (c) Input data of low price
   $$\mu_{1,1}=N\left(h_{3,8} \mid -1,510; 0,719\right) \tag{21}$$
   $$\mu_{1,2}=N\left(h_{3,8} \mid 0,034; 0,780\right) \tag{22}$$
   $$\mu_{1,3}=N\left(h_{3,8} \mid 1,518; 1,160\right) \tag{23}$$
   (d) Input data of close price
   $$\mu_{1,1}=N\left(h_{4,8} \mid -1,490; 1,015\right) \tag{24}$$
   $$\mu_{1,2}=N\left(h_{4,8} \mid 0,050; 1,020\right) \tag{25}$$
   $$\mu_{1,3}=N\left(h_{4,8} \mid 1,540; 0,863\right) \tag{26}$$

Based on equations (15) to (26), can be calculated the fuzzy set membership value of each input data. Figure 6 illustrates a plot showing the calculation of membership values in the close price input data from January 6, 2020.

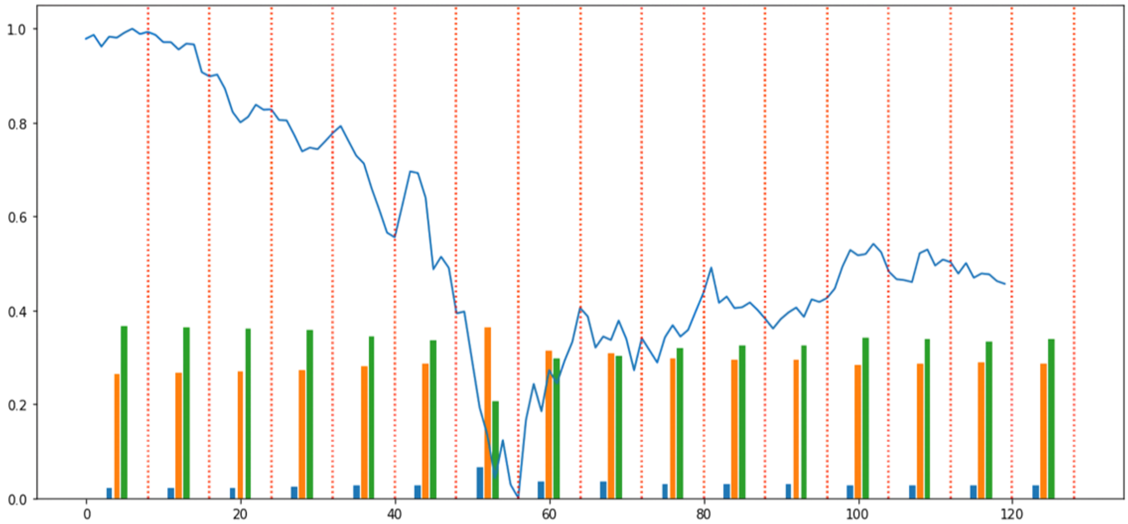

Figure 6. Plot fuzzy membership function of data close price per 8 days

In Figure 6, the blue bar chart is the value of the first fuzzy set member, which has a membership value of less than 0.1. The orange bar chart is the value of the first fuzzy set member, which has a membership value between 0.25 to 0.4. The green bar diagram is the value of the third fuzzy set member, which has a membership value between 0.2 and 0.4.

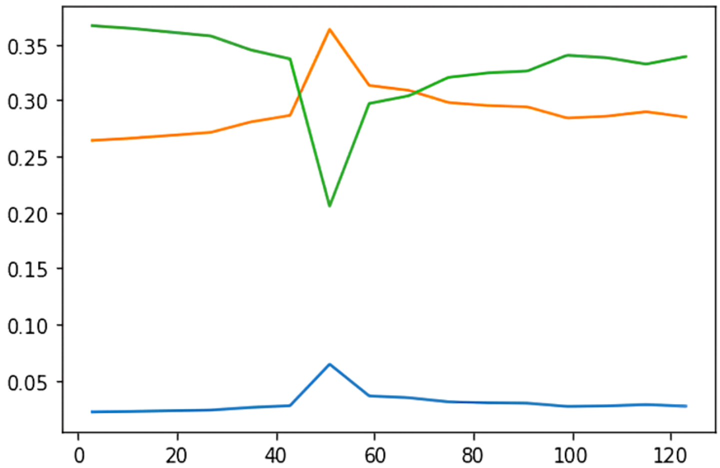

Figure 7. Plot fuzzy membership function of data close price per 8 days

If the values of the members of each fuzzy set are plotted, an interesting pattern will be seen as shown in Figure 7. Membership value of the first fuzzy set (blue color), increases on the 50th day, which on the next day the trend changes from the initial down to the up. Thus it can be said that the first set of membership in the close price data represents an indicator of trend change. The membership value of the second fuzzy set (orange color), also increased on the 50th day like the first fuzzy set. The difference between the first and second fuzzy sets is that in the second fuzzy set there is still movement, both before the

50th day and after. In further, the second fuzzy set appears to more representative of the magnitude of the close price change in the last eight days. Meanwhile, in the membership value of the third fuzzy set (green color), it can be seen that the movement of values every time is very similar to the trends that occur in the data. Thus, it can be said that when compared with the first and second fuzzy sets the third fuzzy set is more representative of the trend pattern at the time.

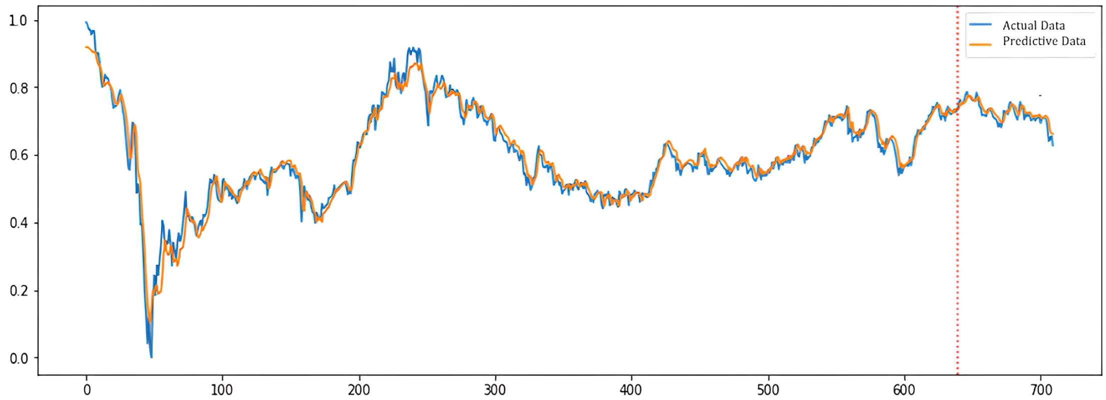

Figure 8. Comparison plot of forecasting data and actual close price

Figure 8 shows the predicted results of RNN-ANFIS. It can be seen that both training data, validation data and test data, RNN-ANFIS is able to forecast very well. Even RNN-ANFIS is still able to recognize quite extreme trend patterns such as on the 50th day.

4. Conclusion

RNN-ANFIS has an excellent ability to forecast JII in the next day. This can be shown from the MSE and MAE values in the last epoch, where the values of the two evaluation metrics have very small values, namely MSE of 0.001 and MAE of 0.246. In addition, this model is also able to explain what indicators are used during the prediction process through fuzzy sets. RNN provides more capabilities to ANFIS so that ANFIS is able to recognize trend patterns in the input data. In the close price input data, the three fuzzy sets represent indicators of trend changes, the magnitude of close price changes, and also patterns in trends.

References

[1] J. A. Goguen, "L. a. zadeh. fuzzy sets. information and control, vol. 8 (1965), pp. 338–353. l. a. zadeh. similarity relations and fuzzy orderings. information sciences, vol. 3 (1971), pp. 177–200." *Journal of Symbolic Logic*, vol. 38, no. 4, pp. 656–657, 1973, doi: 10.2307/2272014.

[2] J. S. R. Jang, "Fuzzy modeling using generalized neural networks and kalman filter algorithm," in *The 9th National Conference on Artificial Intelegence*, 1991, [Online] Available: https://cdn.aaai.org/AAAI/1991/AAAI91-119.pdf.

[3] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.

[4] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985, doi: 10.1109/TSMC.1985.6313399.

[5] A. V. Gite, R. M. Bodade, and B. M. Raut, "Anfis controller and its application," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 2, pp. 1–5, 2013.

[6] G. N. Pillai, J. Pushpak, and M. G. Nisha, "Extreme learning anfis for control applications," in *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*. IEEE, 2014, pp. 1–8, doi: 10.1109/CICA.2014.7013226.

[7] M. R. Prusty, T. Jayanthi, J. Chakraborty, and K. Velusamy, "Feasibility of anfis towards multiclass event classification in pfbr considering dimensionality reduction using pca," *Annals of Nuclear Energy*, vol. 99, pp. 311–320, 2017, doi: 10.1016/j.anucene.2016.09.015.

[8] S. Mahapatra, D. Mohanta, P. K. Mohanty, and S. K. Nayak, "Classification of emg signals using anfis for the detection of neuromuscular disorders," in *Recent Developments in Intelligent Computing, Communication and Devices*, 2017, pp. 53–60, doi: 10.1007/978-981-10-3779-5_8.

[9] I. Svalina, V. Galzina, R. Lujić, and G. Šimunović, "An adaptive network-based fuzzy inference system (anfis) for the forecasting: The case of close price indices," *Expert Systems with Applications*, vol. 40, no. 15, pp. 6055–6063, 2013, doi: 10.1016/j.eswa.2013.05.029.

[10] G. Perveen, M. Rizwan, and N. Goel, "An anfis-based model for solar energy forecasting and its smart grid application," *Engineering Reports*, vol. 1, no. 5, 2019, doi: 10.1002/eng2.12070.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] 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: 10.1016/j.ijforecast.2020.06.008.

[13] M. Farsi, "Application of ensemble rnn deep neural network to the fall detection through iot environment," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 199–211, 2021.

[14] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020, doi: 10.1016/j.physd.2019.132306.

[15] A. Moghar and M. Hamiche, "Stock market prediction using lstm recurrent neural network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020, doi: 10.1016/j.procs.2020.03.049.

[16] M.-Y. Chen, "A hybrid anfis model for business failure prediction utilizing particle swarm optimization and subtractive clustering," *Information Sciences*, vol. 220, pp. 180–195, 2013, doi: 10.1016/j.ins.2011.09.013.

[17] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid cnn-rnn based deep learning approach," *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100007, 2021, doi: 10.1016/j.jjimei.2020.100007.

[18] Z. Liu, Q. Li, J. Zhou, W. Jiao, and X. Wang, "Runoff prediction using a novel hybrid anfis model based on variable screening," *Water Resources Management*, vol. 35, no. 9, pp. 2921–2940, 2021, doi: 10.1007/s11269-021-02878-4.

[19] *Data Jakarta Islamic Index*, 2022, [Online] Available: https://finance.yahoo.com/lookup.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge: MIT press, 2016.

[22] H. Z. Muhammad, *Peramalan Beban Listrik Jangka Panjang Pada Pt. Pln (Persero) Apj Jember Dengan Menggunakan Metode Recurrent Neural Network Dengan Optimasi Levenberg Marquardt*. Universitas Negeri Jember, 2018.

[23] A. G. Salman and Y. L. Prasetio, "Implementasi jaringan syaraf tiruan recurrent menggunakan gradient descent adaptive learning rate and momentum untuk pendugaan curah hujan," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 2, no. 1, pp. 23–35, 2011.

[24] W. Walid, "Peramalan penjualan harga saham pt bank rakyat (persero) tbk bbri indonesia dengan menggunakan recurren neural nerwork (rnn)," in *PRISMA, Prosiding Seminar Nasional Matematika*, vol. 2, 2019, pp. 139–147.

[25] A. Alimuddin, *Teori dan Aplikasi Dasar Sistem Kendali Cerdas*. Serang: Untirta Press, 2020.

[26] S. Sumathi and S. Paneerselvam, *Computational intelligence paradigms: theory & applications using MATLAB*. Florida: Crc Press, 2010.

[27] I. Pakaya, "Particle swarm optimazion-fuzzy logic controller untuk penyearah satu fasa," *Edutic - Scientific Journal of Informatics Education*, vol. 1, no. 1, pp. 1–11, 2015, doi: 10.21107/edutic.v1i1.401.

[28] G. Lenhard, *Adaptive-Network-Based Fuzzy Inference System (ANFIS) based on Keras on top of Tensorflow 2.0.*, 2020, [Online] Available: https://github.com/tiagoCuervo/TensorANFIS.