HASIL CEK_C5 Prosiding ICEAT

by C5 Prosiding Iceat Nursyiva

IOP Conference Series: Materials Science and Engineering

PAPER • **OPEN ACCESS**

Fuzzy feed forward neural network (FFFNN) model for the Jakarta Islamic index (JII) forecasting

To cite this article: N Irsalinda *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **821** 012004

IOP Conf. Series: Materials Science and Engineering **821** (2020) 012004 doi:10.1088/1757-899X/821/1/012004

Fuzzy feed forward neural network (FFFNN) model for the Jakarta Islamic index (JII) forecasting

N Irsalinda*, Y S Astuti and Sugiyarto
Department of Mathematics, Faculty of Science and Applied Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

*Email: nursyiva.irsalinda@math.uad.ac.id

Abstract. Feed Forward Neural Network (FFNN) model is the best model to forecast the time series data. In this research, The Fuzzy Feed Forward Neural Network (FFFNN) with backward propagation method is used to predict the Jakarta Islamic Index (JII) data time series in 2018. Fuzzy is used as input to the FFNN model because it is overcome the weaknesses of the inaccurate results of the FFNN when the data is unclear or incomplete. The purposes of this research are to explain the procedure to generate the FFFNN model. The steps of the FFFNN model prepare the input data to become a fuzzy number using Growth-S Curve fuzzification; the second is divide the data into two training and testing data, the third is determining the best neural network architecture with different neurons and hidden layers to get the best weights that used for the forecasting model. In this research, the best FFFNN model is built by 19 neurons and one hidden layer with 90% and 10% training and testing data, respectively. Therefore with the model obtained, forecasting produces the value of MSE 0.0018 in training and 0.0004 in testing. From the MSE values obtained, it can be concluded that the forecasting using FFFNN model is reasonable to predict.

1. Introduction

Investment is the easiest and intelligent way to accumulate wealth. Financially free people are those who start investing early. They can choose the lifestyle they want by using the investment they have. One of the investment process is by buying shares of companies listed on the stock market. In the investment process, all these investors want to get the most profit and avoid losses. Therefore all investments made need to be planned appropriately. Investors who want to buy company shares need to ensure that the targeted company is a healthy and growing company. That way, the company can generate profits for shareholders through dividends or capital gains. Therefore the investors must do some investment analysis either alone or by using expert analysis services [1].

One of the indices that could be a reference to the condition of the companies listed on the Indonesian stock exchange is the Jakarta Islamic Index (JII). It is established on July 3rd 2000, on the Indonesia Stock Exchange (IDX) (formerly known as Jakarta Stock Exchange) to facilitate the trading of public companies according to Sharia business code [2]. Sharia stocks are securities in the form of shares which are not contrary to Islamic principles in the Capital Market. Therefore for people who want investments without usury and in accordance with Islamic sharia, they can invest in JII. However, to get a significant profit, the investment must be accompanied by an accurate forecasting analysis.

Forecasting is the activity of estimating or predicting what will happen in the future. In this study, forecasting is done to predict the JII index using previous JII indices. The purpose of forecasting is to produce optimum forecasts that do not have errors or as small as possible errors that refer to the Mean Square Error (MSE) forecasts.

One statistical tool commonly chosen in predicting the complex data and having the ability to produce models with high accuracy is Neural Networks. Artificial Neural Network is one of the information processing systems designed to simulate the work of the human brain to solve the problem by learning process through changes in synaptic weight. The neural network model can be divided into two, namely the Feed Forward Neural Network (FFNN) and Recurrent Neural Network (RNN) [3, 4].

FFNN is a neural network model where the learning process goes forward from the input layer to the output layer, while the RNN is a network model that has the characteristics of a feedback connection in the learning process. The algorithm for Artificial Neural Networks operates directly with numbers, so non-numeric data must be converted to numeric data [5].

In Fuzzy Feed Forward Neural Network (FFFNN), the FFNN input network is a fuzzy numbers. The membership value of a fuzzy set is between a closed interval of 0 and 1 [6]. FFNN has been used in some studies as a stock price forecasting tool. Previous research on FFNN, one of which was the FFFNN for forecasting of the Composite Stock Price Index (CSPI) with genetic algorithm variation selection. In this study, the membership function that used is the s-growth fuzzification curve. The activation function in the network architecture is the binary sigmoid function. Then the selection process is roulette wheel selection and rank based on the Genetic Algorithm. Distribution of training data and testing data in this study were 75% and 25%, respectively [7].

Another relevant research is the Application of the Backward propagation Feed Forward Neural Network (FFNN) Method to Predict Share Prices. In this study, the procedure for establishing the FFNN model with the backward propagation algorithm and the results of its application in the weekly period stock data from September 1st 2014 to September 5th 2016. The forecasting results for the period September 13th, 2016 is Rp. 17,405, the difference value is Rp. 45 with Rp. 17,450 as the actual data. The results show that the FFNN method with a backward propagation algorithm is very well used to predict stock prices [8, 9].

From the previous research, we know that the FFNN model is well used to forecast. However, when the data used is incomplete and unclear, it will produce an inaccurate model. To minimizing the inaccuracy of the forecasting model, fuzzy numbers are used in the FFNN input. Fuzzy number generation is done using the growth-S curve fuzzi15fication. This description is the background of this research. The research purposes are to apply the FFFNN and generating the FFFNN model, so that is good to use for forecasting models.

2. Research method

Fuzzy Feed Forward Neural Network (FFFNN) is a neural network with feed-forward type, and input in the form of fuzzy numbers is applied to forecast Jakarta Islamic Index (JII) sharia stock price index with a backward propagation learning algorithm. The forecasting process is explained in the figure below:

IOP Conf. Series: Materials Science and Engineering **821** (2020) 012004 doi:10.1088/1757-899X/821/1/012004

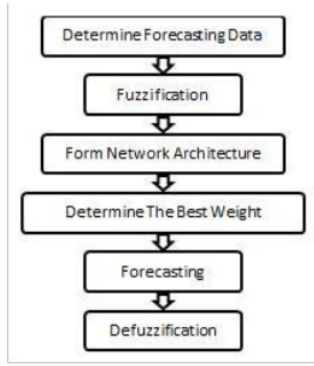

Figure 1. Forecasting process flowchart

The data determined in this research are JII (Jakarta Islamic Index), TASI (Saudi Arabia Stock Price Index), Interest rate, Inflation rate, and SAR (Rupiah's exchange rate to Real Saudi) 172 days in 2018 [6, 7]. The fuzzification is using growth S-curve (Figure 2) and the membership function $\mu_{\tilde{A}(x)}$ obtained by Equation 1 as follows :

$$\mu_{\tilde{A}(x)} = \begin{cases} 0, \; x < a \\ 2\left(\frac{x-a}{b-a}\right)^2, \; a \le x < \frac{(a+b)}{2} \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2, \; \frac{(a+b)}{2} \le x < b \\ 1, \; x \ge b \end{cases}, \quad (1)$$

with x are elements of the decision matrix (research data), a and b are the lower and upper value of the data, respectively.

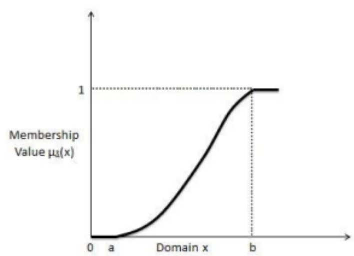

Figure 2. Growth-S Curve

Feed forward neural network (FFNN) is the most frequently used model in the application of time series forecasting [10]. The FFNN model is formulated in Equation 2.

$$y_k = \frac{1}{1+e^{-\left(w_{0k}+\sum_{j=1}^{v}\frac{1}{1+e^{-\left(v_{0j}+\sum_{i=1}^{n}x_i v_{ij}\right)}}\cdot w_{jk}\right)}}, \quad (2)$$

where y_k is k-th output value, v_{0j} = weight of bias hidden layer, v_{ij} = the weight that connecting i-th neuron input to j-th neuron of hidden layer, w_{0k} = weight of bias output layer, w_{jk} = the weight connecting j-th neuron of hidden layer to k-th neuron output, then v and x are the number of hidden layer and the number of input respectively. The architecture of a neural network for forecasting is in Figure 3.

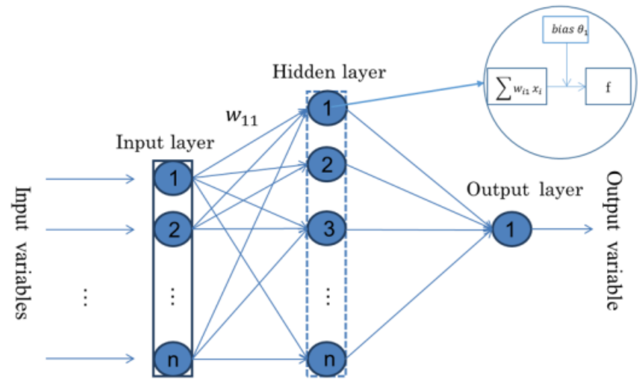

Figure 3. The architecture of neural network used for forecasting (doi: https://doi.org/10.1371/journal.pone.0155133.g001)

To obtain the best model in Equation 2, then we must get the right weights using back propagation algorithm. The backward propagation algorithm is a network learning algorithm with the activation function is binary sigmoid in each layer. This algorithm includes supervised learning algorithms that are often used on feed forward networks. In general, there are three main stages in the back propagation algorithm which is the input process, error tracking and weight adjustment [11, 12]. The steps of the backward propagation algorithm are as follows:

1. Initialize weights conducted randomly or randomly without using scale factors.
2. The feed forward process is used as an algorithm to calculate the activation values that exist in all neurons, both in the hidden layer and the output layer. Following is the algorithm in the feed forward process:
   1) The input layer (x_i with $i = 1,2, ..., m$) received all the input layer neurons are numbered m and forward it to the hidden layer. The input layer is training and testing data.
   2) Hidden layer (z_j with $j = 1,2, ..., n$) calculates all input signals are numbered n by its weight and calculates the activation value. It then forward as an input value for the output layer using the following equation:

$$z(in)_{ij} = v_{0j} + \sum x_i v_{ij} \tag{3}$$

$$z_j = f\big(z(in)_{ij}\big) = \frac{1}{1+e^{-z(in)_{ij}}} \tag{4}$$

The output layer (y_k with $k = 1$) calculates all the hidden layer values with their weights then calculates the activation value as the network output value using the following equation:

$$y(in)_{jk} = w_{0k} + \sum z_j w_{jk}, \tag{5}$$

$$y_k = f\big(y(in)_{jk}\big) = \frac{1}{1+e^{-y(in)_{jk}}}. \tag{6}$$

3. Backward Propagation Process
In this process, a backward calculation is made from the output neurons to the input neurons to produce the appropriate weight value. The output signal generated from the feed-forward process then matched. A difference calculation is made between the target and the output signal

present in the output neuron. Following is the algorithm in the back-propagation process [13-15]:

1) The output layer (y_k with $k = 1$) receives a target pattern that corresponds to the input pattern then calculates k error information using the following equation:

$$\delta_k = (t_k - y_k)\, f'(y(in)_k), \tag{7}$$

t_k is k -th target while y_k is k -th network output. After the error information obtained, then we calculate the difference of weight Δw_{kj} and the correction bias factor Δw_{0j} with the acceleration rate (α) then sent to the hidden layers as follows:

$$\Delta w_{kj} = \alpha\, \delta_k\, z_j, \tag{8}$$

$$\Delta w_{0k} = \alpha\, \delta_k. \tag{9}$$

with δ_k = Correction factor for output to the hidden layer.

2) The hidden layer (z_j with $j = 1,2, \ldots, n$) calculates the hidden units factor $\delta(in)_j$ based on the error in each hidden unit with Equation (10).

$$\delta(in)_j = \sum \delta_k\, w_{kj}. \tag{10}$$

Then $\delta(in)_j$ multiplied by the derivative of the activation function for calculating the error information δ_j as follows:

$$\delta_j = \delta(in)_j\, f'(z(in)_j). \tag{11}$$

Calculating the correction of the difference of weight in the hidden unit Δv_{ji} and the differences of bias factor correction Δv_{0j} with the Equation (12-13) as follows:

$$\Delta v_{ji} = \alpha\, \delta_j\, x_i \tag{12}$$

$$\Delta v_{0j} = \alpha\, \delta_j \tag{13}$$

3) The weights and biases in the output layer and the hidden layer respectively updated with the following equations:

$$w_{kj}(l+1) = w_{kj}(l) + \Delta w_{kj} \tag{14}$$

$$w_{0j}(l+1) = w_{0j}(l) + \Delta w_{0j} \tag{15}$$

$$v_{ji}(l+1) = v_{ji}(l) + \Delta v_{ji} \tag{16}$$

$$v_{0j}(l+1) = v_{0j}(l) + \Delta v_{0j} \tag{17}$$

where l is the previous weight and $l+1$ is a new weight or the updated weight during the iteration.

2.1. Determination of the best network architecture

The data that used in this research are JII (Jakarta Islamic Index) as x_{i1}, TASI (Saudi Arabia Stock Price Index) as x_{i2}, Interest rate as x_{i3}, Inflation rate x_{i4} and SAR (Rupiah's exchange rate to Real Saudi) as x_{i5} and 172 days as i. The data is input in the input layer [16]. The first step is to convert the inputs into fuzzy numbers using Equation (1) so that the data obtained as in Table 1.

Table 1. Fuzzy numbers of input data

No.	$\mu(x_{i1})$	$\mu(x_{i2})$	$\mu(x_{i3})$	$\mu(x_{i4})$	$\mu(x_{i5})$
1.	0.8981	0.0035	0	0.8177	0.0058
2.	0.9256	0.0025	0	0.8177	0.0097
3.	0.9046	0.0008	0	0.8177	0.0130
4.	0.9077	0.0015	0	0.8177	0.0095
5.	0.8879	0.0400	0	0.8177	0.0007
6.	0.8966	0.0626	0	0.8177	0.0009
7.	0.9359	0.0937	0	0.8177	0.0005
8.	0.9513	0.0996	0	0.8177	0.0028
9.	0.9472	0.0812	0	0.8177	0.0009
10.	0.9467	0.0613	0	0.8177	0.0003
11.	0.9850	0.0690	0	0.8177	0.0005
⋮	⋮	⋮	⋮	⋮	⋮
154.	0.0437	0.3835	0.9592	0.5550	0.9993
155.	0.0344	0.4504	0.9592	0.5550	0.9999
156.	0.0598	0.4281	0.9592	0.5550	0.9996
157.	0.0864	0.5452	1	0.7693	0.9982
158.	0.1039	0.4495	1	0.7693	0.9586
159.	0.1049	0.3830	1	0.7693	0.9317
160.	0.1076	0.3975	1	0.7693	0.8750
161.	0.1281	0.3692	1	0.7693	0.8110
162.	0.1039	0.2637	1	0.7693	0.8660
163.	0.0361	0.3450	1	0.7693	0.9330
164.	0.0621	0.3131	1	0.7693	0.8704
165.	0.0805	0.2064	1	0.7693	0.8732
166.	0.2328	0.0708	1	0.7693	0.7645
167.	0.1909	0.0758	1	0.7693	0.7865
168.	0.1503	0.0819	1	0.7693	0.7682
169.	0.1912	0.0808	1	0.7693	0.7393
170.	0.1760	0.0918	1	0.7693	0.7024
171.	0.1358	0.1257	1	0.7693	0.7264
172.	0.0758	0.1787	1	0.7693	0.6225

The next step is generate the network architecture. In this research, the architecture of the neural network is three-layer with 5 neurons in the input layer, trial-error for 11, 13, 15, 17, and 19 neurons in the hidden layer and a neuron in the output layer. The learning method algorithm of this FFFNN is backward propagation with data training and testing 90%:10%, 80%:20%, 70%:30%, 60%:40%, and 50%:50%. The best architecture is that it has a minimum mean square error (MSE) in learning data and testing data [17-19]. MSE can be calculated using this formula:

$$\mathrm{MSE} = \frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2. \tag{18}$$

where Y_t = real value and $\hat{Y}_t$ = forecasting result value [20]. The results of the trial and error from the variation combination network architecture presented in Table 2.

Table 2. Trial-error result for variation of neural network architecture

Training and Testing Data Partition(%)	Hidden Layer Neurons	MSE of Training	MSE of Testing
90:10	11	0.0023	0.0009
	13	0.0022	0.0003
	15	0.0021	0.0008
	17	0.0022	0.0004
	19	*0.0018*	*0.0004*
80:20	11	0.0023	0.0008
	13	0.0024	0.0015
	15	0.0022	0.0006
	17	0.0025	0.0017
	19	0.0024	0.0009
70:30	11	0.0024	0.0167
	13	0.0023	0.0023
	15	0.0021	0.0029
	17	0.0021	0.0015
	19	0.0022	0.0028
60:40	11	0.0027	0.0018
	13	0.0023	0.0038
	15	0.0020	0.1655
	17	0.0021	0.0425
	19	0.0018	0.1216
50:50	11	0.0021	0.0117
	13	0.0022	0.1100
	15	0.0021	0.2991
	17	0.0018	0.0128
	19	0.0023	0.0218

From Table 2, the MSE of training and testing data obtained, the best network architecture is an architecture with five neurons in the input layer, 19 neurons in the hidden layer, and a single neuron in the output layer with 90% training and 10% testing data partition. The following is a comparison chart of FFFNN output with the network architecture obtained:

In Figure 4, the result of the training data using backward propagation (green line) and actual training data (magenta) depicted in a coordinate plane. From the graphs, it appears that there is a slight error of 0.0018. Likewise, in Figure 5, the chart of the testing data using the backward propagation method and testing data actual produces an error of 0.0004. The errors are the minimum error of the best architecture in Table 2. Furthermore, this research obtains the best architecture, such as in Figure 6.

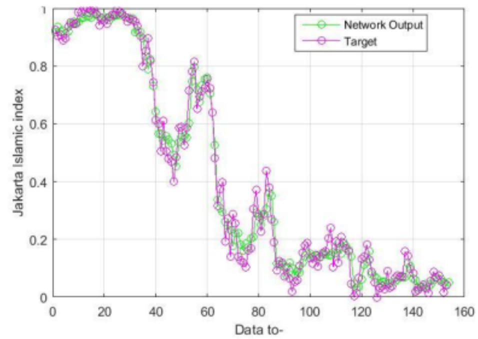

Figure 4. Graphic of backward propagation vs actual training data

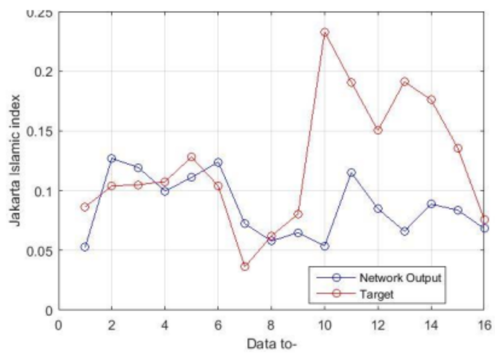

Figure 5. Graphic of backward propagation vs actual testing data

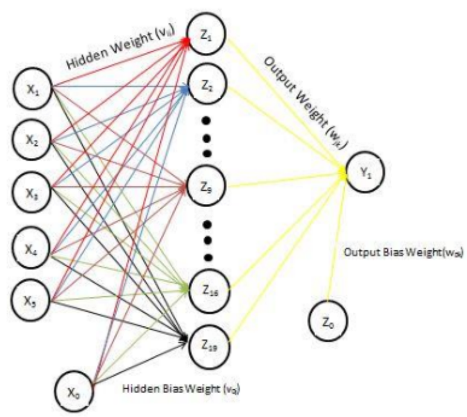

Figure 6. The best architecture in JII forecasting

The best weights in the best network architecture are calculated using Equation (14) until Equation (17). The results of the weights are as follows:

$$v_{ji} = \begin{bmatrix} 5.7936 & 0.9006 & -4.8468 & 1.9951 & -5.9469 \\ -2.8497 & -5.5905 & 6.7140 & -3.3334 & 5.2351 \\ 5.6906 & -1.4511 & 6.4145 & -5.2456 & -0.9282 \\ -6.2568 & -1.5037 & 5.2337 & 5.8911 & 2.9811 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1.3915 & 6.7244 & -5.9223 & 2.8275 & 4.4881 \end{bmatrix}$$

$$\begin{aligned} v_{0j} &= \begin{bmatrix} v_{01} & v_{02} & v_{03} & v_{04} & \cdots & \cdots & v_{019} \end{bmatrix} \\ &= \begin{bmatrix} -7.7416 & 4.6244 & -6.0616 & 1.5558 & \cdots & \cdots & -0.2127 \end{bmatrix} \end{aligned}$$

$$w_{jk} = \begin{bmatrix} w_{11} \\ w_{21} \\ w_{31} \\ w_{41} \\ \vdots \\ \vdots \\ w_{191} \end{bmatrix} = \begin{bmatrix} 2.9894 \\ 2.3722 \\ 1.4600 \\ -2.6154 \\ \vdots \\ \vdots \\ 2.2569 \end{bmatrix}$$

$$w_{0k} = \begin{bmatrix} w_{01} \end{bmatrix} = \begin{bmatrix} 0.0706 \end{bmatrix}$$

3. Result and discussion

From the best architecture and weights obtained, the FFFNN model with backward propagation algorithm in JII stock price index forecasting is as follows:

$$y_k = \frac{1}{1+e^{-\left(w_{0k}+\sum_{j=1}^{19}\frac{1}{1+e^{-\left(v_{0j}+\sum_{i=1}^{5}x_i v_{ij}\right)}}\cdot w_{jk}\right)}}, \tag{19}$$

To forecast day 172 of the JII index, we used data day 171 of JII, TASI, Interest, Inflation and SAR in the form of fuzzy value. JII = 0.1358, TASI = 0.1257, Interest = 1.0000, Inflation = 0.7693 and SAR = 0.7264. After applying these values to FFFNN formed, the result is 0.0683. Then, to get the crisp value of JII, 0.0683 is defuzzification, so that the JII forecasting value is 618.92. Comparison with the real value of the JII index in day 172 is 648.80 have a MSE value of 0.0046.

4. Conclusion

From the discussion, we can conclude that the steps of the FFFNN model prepare the input data to become a fuzzy number using Growth-S Curve fuzzification, the second is divide the data into two training and testing data, the third is determining the best neural network architecture with different neurons and hidden layers to get the best weights that used for forecasting model. In this research, the best FFFNN model is built by 19 neurons and one hidden layer with 90% and 10% training and testing data, respectively. Therefore with the model obtained, forecasting produces the value of MSE 0.0018 in training and 0.0004 in testing. From the MSE values obtained, it can be concluded that the forecasting using FFFNN model is reasonable to predict.

References

[1] Fabozzi F J and Drake P P 2009 *Finance: Capital Markets, Financial Management and Investment Management* (USA: John Wiley).
[2] Tanjung H and Siregar T A 2018 *Journal of Islamic Economics, Finance and Banking* **(1)** pp 147-57.
[3] Dreyfus G 2005 *Neural Networks Methodology and Applications* (Germany: Springer).
[4] Jang J-S R, Sun C T, and Mizutani E 1997 *Neuro-Fuzzy and Soft Computing A Computational Approach to Learning and Machine Intelligence* (Upper Saddle River: Prentice Hall).
[5] Fausset L 1994 *Fundamentals of Neural Networks* (Englewood Cliffs: Prentice Hall).
[6] Herliansyah R and Jamillatuzzahro 2017 *GSTF Journal of Mathematics, Statistics and Operations Research (JMSOR)* **4** (1) pp 8-14.
[7] Deka P and Chandramaoli V 2005 *Journal of Hydrologic Engineering* **3** pp 302-14.
[8] Hansun S 2013 *Ultimatics* **IV** (1) pp 26-30.
[9] Tkacz G 2001 *International Journal of Forecasting* **17** pp 57-69.
[10] Wei W W S 1990 *Time Series Analysis: Univariate and Multivariate Methods* (USA: Addison-Wesley).
[11] Chen X, Racine J, and Swanson N R 2001 *IEEE Transaction on Neural Networks* **12** pp 674-83.
[12] Sudarsono A 2016 J*urnal Media Infotama* **12** 1.
[13] Purushothaman G and Karayiannis N B 2006 *Journal of Applied Functional Analysis* **1** pp 9-32.
[14] Vasant P M, Nagarajan R and Yacoob S 2004 *Scientiae Mathematicae Japonicae* **10** pp 513-27.
[15] Medasani S, Kim J, and Krishnapuram R 1998 *International Journal of Approximate Reasioning* **9** pp 391-417.
[16] Bakar N A, and Rosbi S 2017 *International Journal of Advanced Engineering Research and Science (IJAERS)* **4** pp 130-37.
[17] Alsharif M H, Younes M K and Kim J 2019 *Symmetry* **11** pp 1-17.
[18] Farhath Z A, Arputhamary B and Arockiam D L 2016 *Int. J. Comput. Sci. Mobile Computer* **5** pp 104-9.
[19] Sotirov S and Atanassov K 2009 *Bulgarian Academy of Science Cybernetics and Information Technologies* **9** pp 62-8.
[20] Ullah H and Bhuiyan M A 2018 *Journal of Science and Technology.*

HASIL CEK_C5 Prosiding ICEAT

ORIGINALITY REPORT

8%	8%	10%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Muhammadiyah Purwokerto Student Paper	8%

Exclude quotes	On	Exclude matches	< 2%
Exclude bibliography	On		