Crude Palm Oil Prediction Based on Backpropagation Neural Network Approach

Hijratul Aini [1] and Haviluddin [2, *]

Faculty of Computer Science and Information Technology, Mulawarman University
Jl.Kuaro no.1, Samarinda 75123, Indonesia

[1] hijratulaini10@gmail.com; [2] haviluddin@unmul.ac.id*

* corresponding author

ARTICLE INFO

ABSTRACT

Crude palm oil (CPO) production at PT. Perkebunan Nusantara (PTPN) XIII from January 2015 to January 2018 have been treated. This paper aims to predict CPO production using intelligent algorithms called Backpropagation Neural Network (BPNN). The accuracy of prediction algorithms have been measured by mean square error (MSE). The experiment showed that the best hidden layer architecture (HLA) is 5-10-11-12-13-1 with learning function (LF) of trainlm, activation function (AF) of logsig and purelin, and learning rate (LR) of 0.5. This architecture has a good accuracy with MSE of 0.0643. The results showed that this model can predict CPO production in 2019.

I. Introduction

Indonesia is the largest crude palm oil (CPO) producer in the world. In 2018, Indonesia produced 43 million tons of CPO from 14.03 million hectares plantation. Consequently, it gives significant contribution to the national economy [1][2][3]. CPO production management is very necessary. Therefore, it should be supported with precise estimation based on production data in previous years.

Numerous methods are used in order to obtain accurate prediction results such as statistical methods (i.e., ARMA, ARIMA, SARIMA, and ES) and intelligent computing methods (i.e., fuzzy logic, neural network) [4][5][6][7]. A research by [8] used SARIMA method to predict crude palm oil, in Terengganu, Malaysia. The dataset of CPO and palm kernel from June 2001 until May 2011 was used. The results showed that the SARIMA method was able to predict quite well. Furthermore, [6] have used ANFIS and ARFIMA methods to predict the CPO price in Malaysia. The dataset price CPO from January 2004 until December 2011 was used. The research shows that the ANFIS and ARFIMA models have the ability to use in predicting the CPO prices. On the other hand, [9] have implemented intelligence methods such as Support Vector Machine (SVM), Neural Networks (NN) to predict crude oil prices, palm oil, rubber, and gold. Researchers have confirmed that intelligent algorithms were able to predict accurately compared to statistical method (Random Forest). The prediction results showed that the four parameters greatly affect Malaysia's income. Moreover, [10] have implemented intelligence method, namely Nonlinear Autoregressive with External (NARX) with three algorithms, Levenberg-Marquardt, Bayesian Regulation and Scaled Conjugate Gradient. The research demonstrated that the NARX method was able to predict CPO prices accurately.

This paper aims to apply one of artificial intelligence method, namely backpropagation neural network (BPNN) to predict CPO production. This article consists of fourth sections. Section 1, motivation for writing, research related to what has been prepared. Section 2, the method used for prediction. Section 3, experimental, and section 4 results and discussion then also summary of the study. The analysis results are expected to support management in planning CPO production.

II. Methods

Prediction is an art and science that predicts future events. In other words, the prediction is to require historical data that aims to predict the future. The field of prediction research is increasingly important especially in the economic field. The prediction in production will be required if the conditions of the market are complex and dynamic. Therefore, accurate predictions in assisting management decision making are necessary. Then, numerous algorithms have been existing and developed in the predictions area, from traditional until intelligent algorithms. In this paper, historical data on crude oil palm production have been implemented to be analysed using intelligent algorithms [11][12][13]. This section will briefly explain a predictions, the BPNN algorithm and historical data used.

A backpropagation neural network (BPNN) algorithm is a part of an intelligent method that aims to reduce an error rate in predicting. This method adjust its weight based on the desired output and target differences. The BPNN principle is a multilayer training method by using three layers, namely input layer, hidden layer and output layer, and also weight update process [14].

The BPNN is a development from single layer that only have an input and an output layers. By using a hidden layer, the error value on the network is smaller than the single layer. Where, the hidden layer as a place to update and adjust the weight. Thus, the new weight values are obtained then directed towards the desired output target [14][15]. The BPNN architecture and flowchart can be seen in Figure 1 and Figure 2. The BPNN steps are are listed as follows [13][14][16][17].

- Step 1: Initialize weights with small random value numbers
- Step 2: As long as the stop condition is not fulfilled, do steps 3 to 8
- Step 3: Each input unit receives an xi input signal and is forwarded to hidden units
- Step 4: Each unit is hidden summing the weight of the input signal
- Step 5: each output summing neuron input weight
- Step 6: Each output unit calculated error in each layer
- Step 7: Each hidden layer summing the input layer values from the units in the top layer
- Step 8: Each output calculating update weight and bias
- Step 9: Stop if condition met

A. Feed Forward steps

Feed forward steps contains step 3 to step 5. In step 3, each input unit (x_i, $i = 1, … $ n) receives an xi input signal and is forwarded to hidden units. While in step 4: Each unit is hidden (z_j, $z = 1, ... $ p) summing the weight of the input signal with equation (1):

$$Z\_in_{jk} = v_{0j} + \sum_{n=1}^{i} x_i v_{ij} \tag{1}$$

where Z is neuron hidden, v_{0j} is input bias weight *neuron* to *j*, x_i is neuron input *i*, v_{ij} is neuron input to neuron hidden weights. Applying the activation function was calculated by equation (2):

$$Z_j = f(Z\_in_j) \tag{2}$$

where, Z_j is unit *j* in hidden layer, Z_in_j is unit Z_j output. For example, the activation function used is sigmoid with equation (3):

$$Y = f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The next process is sending all neuron to the output unit where, in step 5, each output (y_k, $k = 1, … $ m) summing neuron input weight using equation (4):

$$Y\_in_k = w_{0k} + \sum_{k=1}^{p} z_j w_{jk} \tag{4}$$

where, Y_in_k is output unit y_k, w_{0k} is weight *bias* for hidden neuron k, z_j is unit *j* in hidden layer, and w_{jk} is hidden neuron to output neuron weights. Finally, applying activation function was using eq (5):

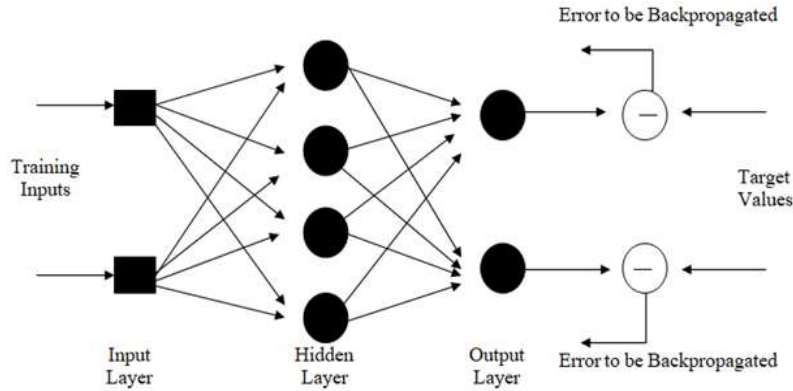$$Y_j = f(Y\_in_k) \tag{5}$$

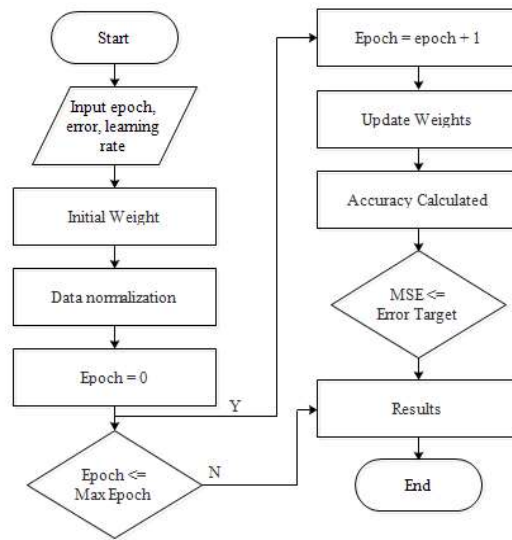where, Y_in_k is unit output y_k

Fig. 1. BPNN architecture

Fig. 2. Flowchart BPNN

B. Backward steps

Feed forward steps contains step 6 to step 7. Step 6 is when each output unit (y_k. k = 1, … m) calculated error in each layer using equation (6).

$$\delta_k = (t_k - y_k)f'(y\_in_k) \tag{6}$$

where, δ_k is weight correction factor w_{jk}, t is target, y_k is output neuron k, y_in_k is output unit y_k. Moreover, weight correction factor and bias was calculated using equation (7).

$$\Delta w_{jk} = \alpha \delta_k x_j \tag{7}$$

$$\Delta w_{0k} = \alpha \delta_k$$

where, Δw_{jk} is hidden neuron weight w_{jk} (t) with $w_{jk\ (t+1)}$, Δw_{0k} is bias weight for hidden neuron k, a is learning rate, δ_k is weight correction factor w_{jk}, x is input

In step 7, Each hidden layer (Z_j, $Z = 1$, … p) summing the input layer values (a) from the units in the top layer, by using equation (8).

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \tag{8}$$

where, δ_k is weight correction factor w_{jk}, w_{jk} is neuron hidden to neuron output.

In the next process, calculating error in each layer has been done using equation (9).

$$\delta_j = \delta\_in_j f(x\_in_j) \tag{9}$$

where, δ_j is weight correction factor v_{ij}, δ is correction factor, x is input. While calculating weight correction factor and bias has been done using equation (10).

$$\Delta v_{ijj} = \alpha \delta_j x_i \tag{10}$$

where, Δv_{ij} is neuron input weight to neuron hidden, α is learning rate, δ_j is weight correction factor v_{ij}, x_i is neuron input i

C. Update weight and bias steps

Feed forward steps contains step 8 where each output (y_k, k = 1, … m) calculating update weight and bias (j = 0.1, ... p) using equation (11).

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \tag{11}$$

where, w_{jk} is neuron hidden to neuron output weights, Δw_{jk} is difference in weight of hidden neurons to output neurons

Each hidden layer (z_j, z = 1, … p) calculating update weight and bias (i = 0.1, ... n) using equation (12).

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \tag{12}$$

where, v_{ij} is neuron input to neuron hidden weights, Δv_{ij} is difference in weight of hidden neurons to output neurons

In this study, historical data was obtained from PT. Perkebunan Nusantara XIII Long Pinang Village, Paser, East Kalimantan, Indonesia. The Fresh Fruit Bunch (FFB) harvest data from 2015 to 2018. A normalized FFB data can be seen in Table 1

The algorithm performance must be measured. A statistical methods (i.e., SSE, R, R2, MAPE, MSE etc.) are usually used to measure the algorithm performance [7]. In this paper, Mean Squared Error (MSE) has been performed to evaluate the BPNN algorithm in predictions. The MSE are sum of squares for all prediction errors values in each period and dividing by the number of prediction periods [18][19][20]. The MSE can be calculate using (1), where i is real values, and I is predicted values

$$MSE = \frac{1}{n} \sum_{i-1}^{n} (Y_i - \hat{Y}_i)^2 \tag{13}$$

where, Y_i is real values, and $\hat{Y}_i$ is predicted values

Table 1. Harvest data of TBS Inti Tajati (2015–2018) after normalization

Years/Months	2015	2016	2017	2018
January	0.4460	0.5143	0.5052	0.5010
February	0.3510	0.3315	0.5960	0.3398
March	0.3695	0.2659	0.4005	0.1143
April	0.4249	0.1523	0.3348	0.1000
May	0.5156	0.1256	0.3207	0.1501
June	0.5862	0.2659	0.3163	0.1000
July	0.6061	0.2724	0.5103	0.1000
August	0.5214	0.4077	0.1000	0.1930
September	0.7185	0.6322	0.5582	0.6115
October	0.9000	0.7779	0.7613	0.7360
November	0.8830	0.8298	0.7613	0.1000
December	0.7977	0.7015	0.6097	0.1000

III. Results and Discussion

In this experiment, the crude palm oil prediction have been tested to get a good BPNN model. Therefore, a try-and-error approach has been implemented. Several variables includes hidden layer architecture (HLA), learning function (LF), activation function (AF) and learning rate (LR) have been explored. Furthermore, the BPNN variables can be seen in Table 2.

Based on neural network principles, a total of 48 data have divided by 2 include 36 for training and 12 for testing data. Meanwhile, five inputs data (2015, 2016, 2017, and 2018) with bias and one output have been utilized.

Based on the experiment, the hidden layer architectures (HLA) were 5-10-11-1 (2); 5-10-11-12-13-1 (4); 5-10-11-11-12-12-13-1 (6); 5-10-11-11-12-12-12-13-13-1 (8).The learning function (LF) were trainlm; traingd; traingdx. The activation function (AF) on the input and hidden layers were logsig, and the output layer was purelin. The learning rate (LR) were 0.1; 0.3; 0.5; 0.7 and also the other variables values, maximum epoch of 1.000, and error limit of 0.01 have been evaluated. Afterwards, mean square error (MSE) is used to statistically measure the forecasting accuracy. In principle, the best BPNN architecture and variables is one with the lowest MSE value. Table 3 shows the results of BPNN training and testing.

After experiment many times, the BPNN architectures by using hidden layer architectures (HLA) 5-10-11-12-13-1 (4), the learning function (LF) was trainlm, activation function (AF) on the input and hidden the layers were logsig, and the output layer was purelin, learning rate (LR) were 0.5 and 0.7, maximum epoch of 1.000, and the error limit of 0.01 have been produced good model with MSE of 0.0643. The result can be seen in Table 4.

Based on the best BPNN parameters (in Table 4), the forecasting for the following year has been carried out. In this test, the first BPNN model (with LR=0.5) have been used for predict the next year (2019) production. Figure 3 and Figure 4, the first BPNN model results of training and testing have almost the same values with the target.

Table 2. BPNN Parameters

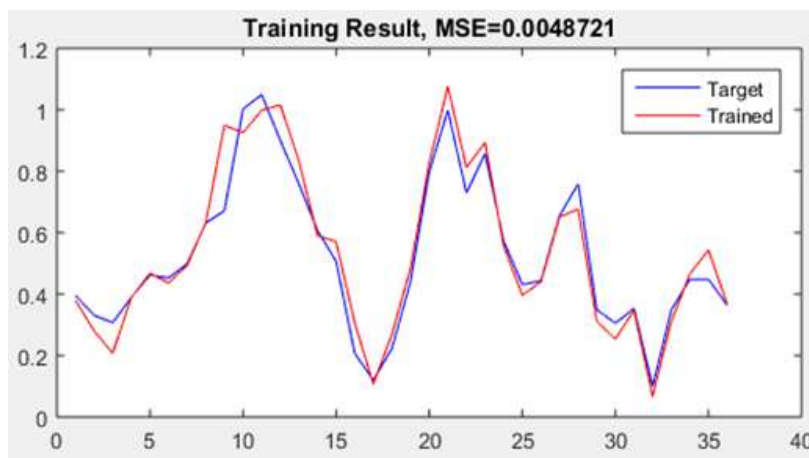| Variables	Values
Hidden layer architecture (HLA)	2, 4, 6, 8 5-10-11-1; 5-10-11-12-13-1; 5-10-11-11-12-12-13-1; 5-10-11-11-12-12-12-13-13-1
Learning function (LF)	trainlm; traingd; traingdx
Activation function (AF)	logsig; purelin
Learning rate (LR)	0.1; 0.3; 0.5; 0.7

Fig. 3. Plot results of training BPNN with LR 0.5

Table 3. BPNN training and testing results

| No	TBS Inti Tajati				
	Parameters			**MSE**	
	Hidden Layer	**Training Function (TF)**	**Learning Rate (LR)**	**Training**	**Testing**
1		trainlm	0.1	0.0075	0.0962
2		traingd	0.1	0.0272	0.0173
3		traingdx	0.1	0.0311	0.1079
4		trainlm	0.3	0.0030	0.1291
5		traingd	0.3	0.0231	0.3381
6	2	traingdx	0.3	1.7055	0.0873
7		trainlm	0.5	0.0029	0.1088
8		traingd	0.5	1.0789	0.1290
9		traingdx	0.5	0.0349	0.1596
10		trainlm	0.7	0.0093	0.0431
11		traingd	0.7	0.7285	0.0706
12		traingdx	0.7	0.0182	0.1100
1		trainlm	0.1	0.0087	0.0091
2		traingd	0.1	0.0119	0.0446
3		traingdx	0.1	0.0081	0.0240
4		trainlm	0.3	0.0035	0.1237
5		traingd	0.3	0.0089	0.0952
6	4	traingdx	0.3	0.0174	0.0806
7		trainlm	0.5	0.0147	0.0033
8		traingd	0.5	0.0099	0.0382
9		traingdx	0.5	0.0194	0.2411
10		trainlm	0.7	0.0072	0.0015
11		traingd	0.7	0.0158	0.1128
12		traingdx	0.7	0.0380	0.1350
1		trainlm	0.1	0.0071	0.0211
2		traingd	0.1	0.0148	0.0601
3		traingdx	0.1	0.0431	0.0673
4		trainlm	0.3	0.0184	0.1616
5		traingd	0.3	0.0283	0.0580
6	6	traingdx	0.3	0.0420	0.0943
7		trainlm	0.5	0.0118	0.1586
8		traingd	0.5	0.0195	0.0809
9		traingdx	0.5	0.0990	0.0521
10		trainlm	0.7	0.0127	0.1101
11		traingd	0.7	0.0455	0.1083
12		traingdx	0.7	0.0103	0.1288
1		trainlm	0.1	0.0368	0.0467
2		traingd	0.1	0.0455	0.1045
3		traingdx	0.1	0.0658	0.0331
4		trainlm	0.3	0.0078	0.6451
5		traingd	0.3	0.6687	1.6575
6	8	traingdx	0.3	0.0103	0.0463
7		trainlm	0.5	0.0191	0.1633
8		traingd	0.5	0.4928	0.8608
9		traingdx	0.5	0.0455	0.1114
10		trainlm	0.7	0.0192	0.0425
11		traingd	0.7	0.8493	0.3114
12		traingdx	0.7	0.0473	0.0883
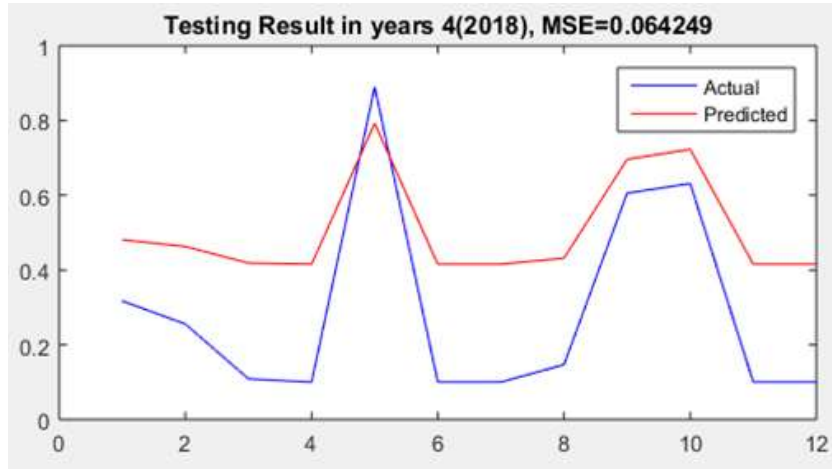
Fig. 4. Plot of testing BPNN with LR 0.5
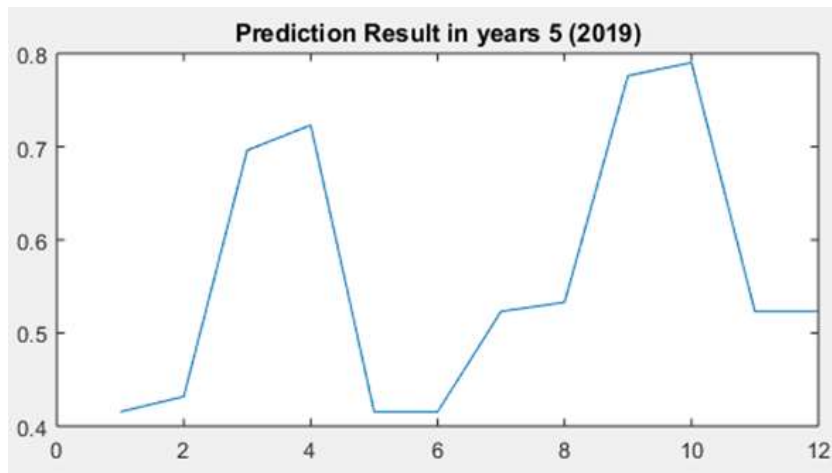
Fig. 5. CPO Prediction Year 2019

Table 4. The best parameters

Farm	Hidden Layer	Training Function (TF)	Learning Rate (LR)	Architecture	Mean Square Error (MSE)	
					Training	Testing
TBS Inti	4	trainlm;	0.5	5-10-11-12-13-1	0.0049	0.0643
Tajati	4	purelin	0.7		0.0214	0.0652

Furthermore, the CPO prediction in January – December 2019 has been carried out with the first BPNN architecture. Figure 5 shows the CPO prediction. The prediction slowly increased until April, then dropped in May and June. Afterwards, it mat increased until October and decreased again until December.

Table 5 shows the prediction results based on the first BPNN model. The average prediction is 1668008.93 with -0.175 of error. It can be concluded from the table that there are movements in monthly CPO production. In Table 5, prediction results based on the first BPNN model. Based on experiment, the average prediction of 1668008.93 and error prediction of -0.175 have been obtained. In other words, in each month there are an increase and decrease in CPO production.

Table 5. Prediction results TBS Inti Tajati

2019	Prediction	Actual	Difference	Error
January	2232026.331	1983190	-248836.3	-0.0503
February	1468812.350	1185890	-282922.3	-0.0572
March	1285529.301	70830	-1214699	-0.2456
April	1312788.635	0	-1312789	-0.2654
May	1237399.359	247820	-989579.4	-0.2001
June	1312788.635	0	-1312789	-0.2654
July	1312788.635	0	-1312789	-0.2654
August	1217143.764	459980	-757163.8	-0.1531
September	2771911.583	2529800	-242111.6	-0.0489
October	3239341.295	3145660	-93681.29	-0.0189
November	1312788.635	0	-1312789	-0.2654
December	1312788.635	0	-1312789	-0.2654
Total	20016107.16	9623170	-10392937	-2.1012
Average	1668008.93	801930.833	-866078.1	-0.1751

IV. Conclusion

The implementation of backpropagation neural network (BPNN) method has been presented. In this study, several variables values (i.e., hidden layer architecture (HLA), learning function (LF), activation function (AF) and learning rate (LR) and other parameter values such as maximum epoch and error limit) have been investigated. Based on experiment, BPNN architecture with 5-10-11-12-13-1, learning rate of 0.5, learning function of trainlm, and activation function of logsig and purelin has a very good accuracy with mean square error (MSE) of 0.064249. Therefore, this model can be used to predict crude palm oil production in 2019. The BPNN with metaheuristic optimization will be conducted in the future experiment.

Acknowledgment

References

[1] Kementan, "Kementan: Industri Kelapa Sawit Berkontribusi Besar Terhadap Ekonomi," *KOMPAS.com*, 2018.

[2] B. P. Statistik, "Badan Pusat Statistik," 2018. [Online]. Available: https://bps.go.id/subject/6/tenaga-kerja.html#subjekViewTab3. [Accessed: 27-Aug-2018].

[3] A. Norhidayu, M. Nur-Syazwani, R. Radzil, I. Amin, and N. Balu, "The production of crude palm oil in Malaysia," *Int. J. Econ. Manag.*, vol. 11, no. 3 Special Issue, pp. 591–606, 2017.

[4] Haviluddin and N. Dengen, "Comparison of SARIMA, NARX and BPNN models in forecasting time series data of network traffic," in *Proceeding - 2016 2nd International Conference on Science in Information Technology, ICSITech 2016: Information Science for Green Society and Environment*, 2017.

[5] H. Haviluddin and A. Jawahir, "Comparing of ARIMA and RBFNN for short-term forecasting," *Int. J. Adv. Intell. Informatics*, vol. 1, no. 1, pp. 15–22, 2015.

[6] A. A. Karia, I. Bujang, and I. Ahmad, "Forecasting on Crude Palm Oil Prices Using Artificial Intelligence Approaches," *Am. J. Oper. Res.*, 2013.

[7] Purnawansyah, Haviluddin, R. Alfred, and A. F. O. Gaffar, "Network Traffic Time Series Performance Analysis using Statistical Methods," *Knowl. Eng. Data Sci.*, vol. 1, no. 1, pp. 1–7, 2018.

[8] S. Ahmad and H. A. Latif, "Forecasting on the Crude Palm Oil and Kernel Palm Production: Seasonal ARIMA Approach," in *2011 IEEE Colloquium on Humanities, Science and Engineering Research (CHUSER 2011), Dec 5-6 2011, Penang*, 2011, pp. 939–944.

[9] S. Ramakrishnan, S. Butt, M. A. Chohan, and H. Ahmad, "Forecasting Malaysian exchange rate using machine learning techniques based on commodities prices," in *International Conference on Research and Innovation in Information Systems, ICRIIS*, 2017.

[10] D.H. Arasim, A.A. Karia, "Identifying and Forecasting the Factors that Derive CPO Prices in Malaysia Using NARX Model," *Int. J. of Case Studies*, Vol. 4, No. 2, pp. 04-14, 2015.

[11] M. Geurts, G. E. P. Box, and G. M. Jenkins, "Time Series Analysis: Forecasting and Control," *J. Mark. Res.*, 2006.

[12] L. Seymour, P. J. Brockwell, and R. A. Davis, "Introduction to Time Series and Forecasting," *J. Am. Stat. Assoc.*, 2006.

[13] Z.-Y. Wang, Y.-C. Lin, S.-J. Lee, and C.-C. Lai, "A Time Series Forecasting Method," *ITM Web Conf.*, 2017.

[14] R. Rojas, "The Backpropagation Algorithm," in *Neural Networks*, 2011.

[15] Purnawansyah and H. Haviluddin, "Comparing performance of Backpropagation and RBF neural network models for predicting daily network traffic," in *2014 Makassar International Conference on Electrical Engineering and Infonnatics (MICEEI)*, 2014, pp. 166–169.

[16] Haviluddin and R. Alfred, "A genetic-based backpropagation neural network for forecasting in time-series data," in *Proceedings - 2015 International Conference on Science in Information Technology: Big Data Spectrum for Future Information Economy, ICSITech 2015*, 2016.

[17] M. Lehtokangas, "Modelling with constructive backpropagation," *Neural Networks*, 1999.

[18] B. K. Nelson, "Statistical methodology: V. Time series analysis using autoregressive integrated moving average (ARIMA) models," *Acad. Emerg. Med.*, 1998.

[19] A. Kaya, "Statistic Modelling for Outlier Factors," 2010.

[20] J. Fürnkranz *et al.*, "Mean Squared Error," in *Encyclopedia of Machine Learning*, 2010.