Use of Empirical Mode Decomposition in Improving Neural Network Forecasting of Paddy Price

[1]Siti Nabilah Syuhada Abdullah, [2]Ani Shabri* and [3]Ruhaidah Samsudin

[1,2]Department of Mathematical Sciences, Faculty of Science, UniversitiTeknologi Malaysia
81310 UTM Johor Bahru, Malaysia

[3]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Malaysia

*Corresponding author: ani@utm.my

Abstract Since rice is a staple food in Malaysia, its price fluctuations pose risks to the producers, suppliers and consumers. Hence, an accurate prediction of paddy price is essential to aid the planning and decision-making in related organizations. The artificial neural network (ANN) has been widely used as a promising method for time series forecasting. In this paper, the effectiveness of integrating empirical mode decomposition (EMD) into an ANN model to forecast paddy price is investigated. The hybrid method is applied on a series of monthly paddy prices from February 1999 up to May 2018 as recorded in the Malaysian Ringgit (MYR) per metric tons. The performance of the simple ANN model and the EMD-ANN model was measured and compared based on their root mean squared Error (RMSE), mean absolute error (MAE) and mean percentage error (MPE). This study finds that the integration of EMD into the neural network model improves the forecasting capabilities. The use of EMD in the ANN model made the forecast errors reduced significantly, and the RMSE was reduced by 0.012, MAE by 0.0002 and MPE by 0.0448.

Keywords Time series forecasting; ANN; EMD; hybrid model; univariate data.

Mathematics Subject Classification 62-04, 62-07, 62M10, 93E11, 62P20, 91B84.

1 Introduction

Rice is considered a staple food in Malaysia, but Malaysia's rice self-sufficiency level (SSL) is currently rising from 65% to 75% following the Third Agricultural Policy (1998-2010) to meet at least 70% of Malaysia's demand. The remaining 30% was supplied from Thailand, Vietnam and China [1].The rise in self-sufficiency of our national rice has become a strategic concern in the Ministry of Agriculture to achieve 100% self-sufficiency in rice [2].

The ability to forecast the future price helps avoid fluctuations that would either burden the vast consumers or eschew the losses incurred by producers and distributors of rice. The risk in the rice price is relatively large because of its seasonal production and high dependency on the weather and it is confronted with a relatively fixed demand and inelastic inter-temporal price changes [3]. Rice prices should reflect the interests of producers and consumers. Reducing price uncertainty can provide the support for the development of food crops. This will determine the expectations of farmer producers, traders, and other economic actors.

It is a daunting task to forecast financial data. Since market imperfections are quickly discovered, manipulated and resolved by market participants, future prices are difficult to predict. Nonetheless, financial time series forecasting is a very interesting subject with applications ranging from trading strategies to risk management [4-10].

The artificial neural networks (ANN) has become increasingly popular in forecasting philosophy, leading to effective implementations in the prediction of time series and explanatory forecasting [11]. Notwithstanding its theoretical capabilities, ANN has been unable to confirm its ability in predicting performance against existing statistical methods, such as the autoregressive integrated moving average (ARIMA) or exponential smoothing [12]. ANN provides several degrees of freedom in the modeling process which include availability of activation functions, sufficient input selection, hidden and output nodes and learning algorithms which include their scientifically sound use is frequently considered as much as an art as science is [13].

Empirical Mode Decomposition (EMD) introduced by [14], manages data non-stationarity effectively in order to break down the initial financial time series up into multiple elements called intrinsic mode functions (IMFs). That extracted IMF contains narrow-range oscillatory patterns that can be interpreted as a quasi-stationary portion. Widespread practice suggests using each IMF as an independent time series with its own model for predicting machine learning [15]. The applications of EMD for forecasting purposes in various fields such as oil and gas [16], exchange rates [17], stock indices [18], sales [19], and demand [20] prove that in reality, this time-scale decomposition is an efficient method with the "divide-and-conquer" theory in mind [9].

The primary aim of this paper is to highlight how, particularly when used with ANN, EMD enhances the predictive capabilities. Using performance metrics such as the root mean squared error (RMSE), mean absolute error (MAE) and mean percentage error (MPE), the hybrid model comprising EMD and ANN is compared to a simple ANN model.

The remainder of this paper is as follows. Section 2 provides a detailed description of the technique. This is accompanied by a review of findings in Section 3 and conclusions and suggestions set out in Section 4.

2 Methodology

This study uses the monthly rice prices (MYR per metric ton) collected from www.indexmundi.com from February 1999 to May 2018.A total of 232 data points was used. As common practice, the data was partitioned into two parts. The first 80% of data was used for training purposes and the following 20% for testing[3-4]. The performance criteria calculated for the training part are used in selecting the best architecture of ANN. Meanwhile,

the performance of the testing part is used to compare and show the improvements of forecasting abilities in the ANN model once it is integrated with EMD.

2.1 Artificial Neural Network (ANN)

Overall, ANN has numerous attributes that make them effective in solving complicated issues. ANN's major advantage is its ability to model a dynamic nonlinear model. ANN has been commonly used in many ways over the recent decades, including financial forecasts [21–24]. The use of ANN can have an extremely complex architectures from selecting various activation functions, number of network layers, number of hidden nodes, right to the selection of specific lags as input data. For different approximation capabilities, each can lead to different models. The simplest multi-layer perceptron with a 3-layer was used in this article. Namely, the input, hidden and output layer.

The ANN model carries out a non-linear approximation of observed values as inputs $(y_{t-1}, y_{t-2}, ..., y_{t-p})$ to better predict values as output (y_t), i.e.,

$$zy_t = a_0 + \sum_{j=1}^{q} a_j f\left(w_{0j} + \sum_{i=1}^{p} w_{ij} y_{t-i}\right) \varepsilon_t, \tag{1}$$

where, $t = 1, 2, ..., n$ is the index of time for a data series, $a_j, (j = 0, 1, 2, ..., q)$ is a bias on the jth unit and $w_{ij}, (i = 0, 1, 2, ..., p; j = 0, 1, 2, ..., q)$ is the connection weights between model layers, p is the number of input nodes, q is the number of hidden nodes and $f(\cdot)$ is the transfer function of the hidden layer in which the study used the logistic sigmoid function [25] defined as

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{2}$$

Discussed further are the methods used to select thef input and tuning the number of nodes in the hidden layer

2.1.1 Input Selection

Several studies have indicated that input nodes specification might just be the most influential factor in the design of an effective multi-layer perceptron (MLP)[26-29]. This is because it provides vital information on the complex autocorrelation structure (linear and/or nonlinear) in the data [30]. The heuristic method of selecting Inputs is used in this study [30]. It follows these three rules:

- 1st rule: Using all time lags from 1 to a given limit of $m :< 1, 2, \ldots, m >$ by evaluating the partial autocorrelation (PACF) function.

- 2nd rule: Choose 4 lags with the highest value of autocorrelation (ACF) function.

- 3rd rule: Based on these decomposition information

  (a) $< 1, K, K + 1 >$ if the series shows trend with seasonal (period K)properties;
  (b) $< 1, K >$ in the absence of trend but shows seasonal (period K) properties in the series; and
  (c) $< 1 >$ and $< 1, 2 >$ if the series is trended without seasonal.

2.1.2 Hidden Layer

The hidden neuron can affect the error on the nodes that are connected to their output. The neural network's stability is determined by error. The reduced error indicates increased stability, and higher error indicates lower stability. Over fitting might be caused by over estimation of hidden neurons; that is, the neural networks exaggerate the difficulty of the target problem [31].

A classic way is to choose a number between the number of input and output neurons. [32]. In addition, Kolmogorov's theorem states that any function of i variables can be represented by superimposing a set of $2i + 1$ univariate function to derive the arbitrary cap for setting the number of hidden neurons, i being the number of inputs for a single-layer ANN model. [33].

2.2 Empirical Mode Decomposition (EMD)

EMD essentially decomposes a complex signal into a low number of underlying intrinsic mode functions (IMFs), organized from increasing frequencies, based on the local characteristic scale, which is described as the lenghth between two successive local extremes. An IMF is a structure that has symmetrical upper and lower envelopes. In addition, the number of zero intersections and the number of extremes are equal or different, at most one [14]. Computed IMF comprises narrow-range oscillatory scales and is generally viewed as a quasi-stationary portion. For example, the seasonal aspect can be interpreted as an IMF extracted from an economic time series with a scale of three months. The IMFs can be derived from the time series data set as outlined in the following sequence through an iterative decomposition process [1]:

- <u>Step 1</u>: All the local time series extremes y_t for $t = 1, 2, \ldots, n$ must be identified;

- <u>Step 2</u>: Link all the local maxima as the upper envelope with a cubic spline line $e_{\max}(t)$ and repeat the process to generate the lower envelope for the local minima $e_{\min}(t)$

- <u>Step 3</u>: Compute the upper and lower mean envelopes based on the formula:

$$m_t = \frac{e_{\max}(t) + e_{\min}(t)}{2} \tag{3}$$

- <u>Step 4</u>: Harvest from the data all mean and describe the difference as $d_t = y_t - m_t$

- <u>Step 5</u>: Ensure d_t satisfies two conditions of IMF, output it as the ith IMF and substitute y_t with the residue $r_t = y_t - d_t$. If d_t is not an IMF, substitute y_t with d_t

- <u>Step 6</u>: Until the residue becomes a fixed function, or a function with only one maximum and one minimum from which IMF can no longer be removed.

  Steps 1 to 5 are to be repeated continuously.

Backward engineering can be applied to obtain the original time series y_t expressed as the sum of these IMFs and as a residue which is given by

$$y_t = \sum_{i=1}^{N} imf_i(t) + r_t, \tag{4}$$

where N is the number of IMFs, and r_t is the final residue.

2.3 EMD-ANN

EMD-ANN is essentially a combination of EMD and ANN to forecast a set of data from time series. First the initial set of data is broken down into IMFs and residue.

Secondly, the ANN approach is applied to each IMF. It begins by building the architecture based on the Kolmogorov theorem for the hidden nodes and the input selection using the heuristic Method. The inputs will then be standardized, partitioned and trained for learning before the model evaluation process is predicted. The weights are revised and modified via learning

The produced values are scaled back to the original range before they are summed up across all IMFs to produce the final forecast values. The procedure flow is as illustrated in Figure 1.

2.4 Performance Criteria

2.4.1 Root Mean Square Error (RMSE)

The root-mean-squared error (RMSE) is a widely used estimate of the variations between the expected values of a model and the actual values. The RMSE is the square root of the differences between the values predicted and observed. These deviations are called residuals when calculations are performed over the estimated data and when calculated out of sample are called errors. RMSE will always be positive, and a value of zero (almost never accomplished) would imply a perfect fit for the test. A lower RMSE is generally better than a higher one

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t)^2}. \tag{5}$$

2.4.2 Mean Absolute Error (MAE)

The mean absolute error (MAE) is an average of the absolute errors between the prediction and the actual value. It is also the average vertical distance between each point and the identity line.

$$MAE = \frac{1}{n} \sum_{t=1}^{n} |\hat{y}_t - y_t|. \tag{6}$$

2.4.3 Mean Percentage Error (MPE)

Mean percentage error (MPE) is the estimated average of an expected value of the percentage error from a prediction that is different from the actual quantity predicted values.Since the real rather than absolute values of the forecast errors are used in the formula, both positive and negative forecast errors can account for each other; as a consequence, the formula can be used as an estimate of the prediction bias. When predictions are consistently low, the resulting value will be positive, indicating the model inadequacy, while if consistently high, it will be negative, indicating over fitting

$$MPE = \frac{100\%}{n} \sum_{t=1}^{n} \frac{y_t - \hat{y}_t}{y_t}. \tag{7}$$
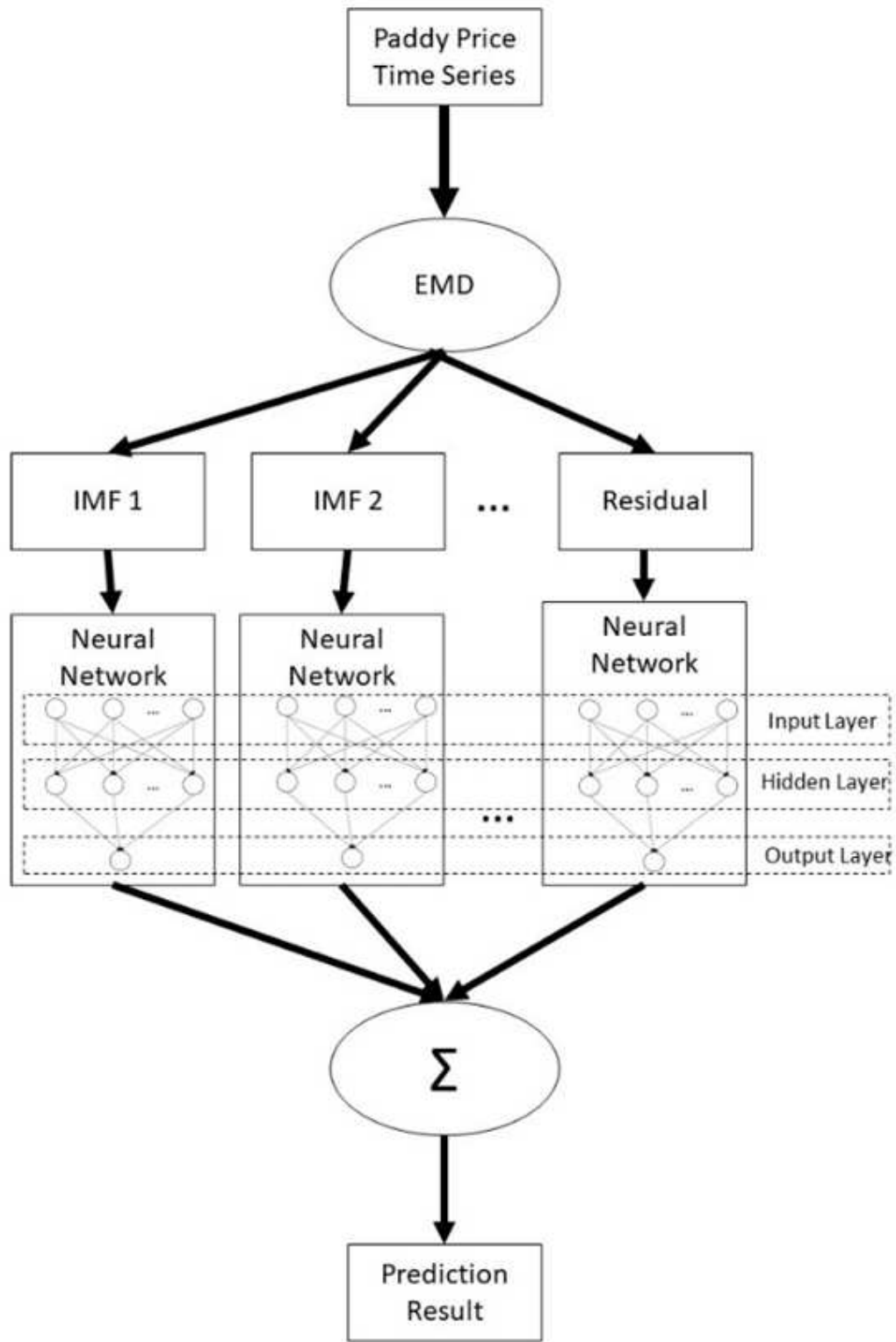
Figure 1: EMD-ANN Hybrid Procedure

3   Results and Discussion

3.1   Artificial Neural Network (ANN)

The plots in Figure 2 show the ACF and PACF of the time-series data. Following the heuristic method, four possible inputs were considered for the architecture. By analyzing the PACF diagram, m was set to 3 based on the 1st Rule. Therefore, all time lags between 1 and a given limit of $m = 3$ were used: $< 1, 2, 3 >$. Next, based on the 2nd rule: the first four lags which had the highest ACF value $< 1, 2, 3, 4 >$ were chosen. Finally, referring to the 3rd rule, since the series is trended non-seasonally, another two architectures $< 1 >$ and $< 1, 2 >$ are considered.
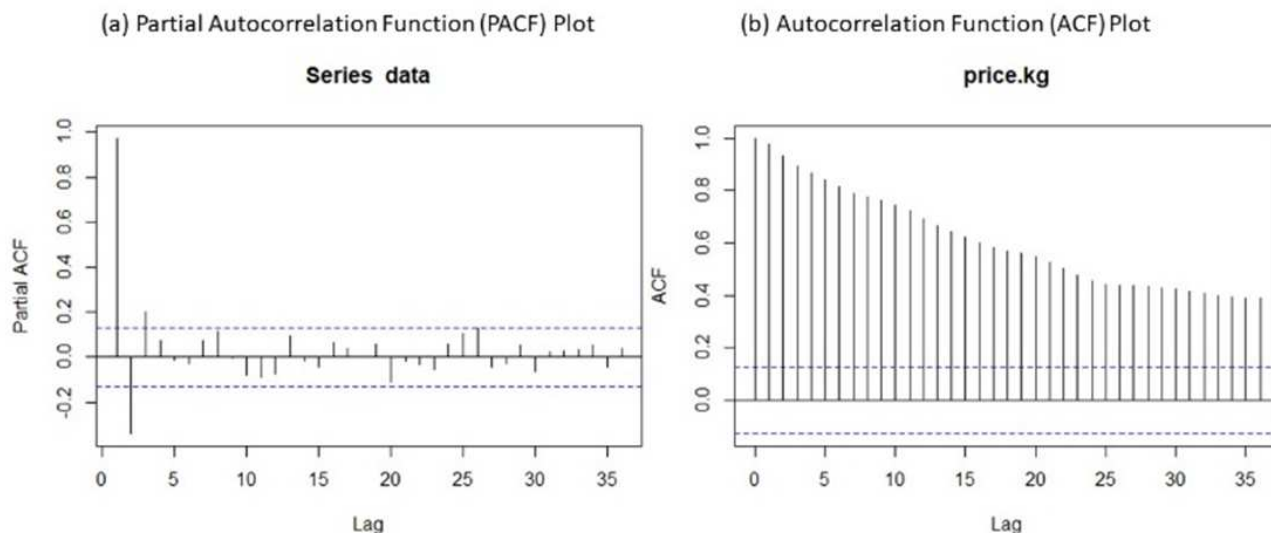

Figure 2: Partial Autocorrelation and Autocorrelation Function of Full Data

Kolmogorov's theorem of selecting hidden nodes [33] was applied to all four possible architecture types. An experimentation approach to optimize the number of neurons in the hidden layer was taken based on the theorem. For example, based on the 1st Rule, an architecture of $< 1, 2, 3 >$ was considered. Hence, based on [33], the study ran the data multiple times, each run using a different number of hidden nodes ranging from 1 until 7 hidden nodes. The number of nodes which yields the lowest error measures for each architecture considered are then compared for the final selection of the ANN model.

Each network was trained at a learning rate of 0.001 for 5000 epochs using the back propagation algorithm. The network that yielded the best training result was selected as the best ANN for the corresponding series. The experiment was repeated ten times and afterwards the error measures were computed. Table 1 demonstrates four ANN architectures output with optimized hidden nodes.

Conclusively, the best simple ANN model was chosen based on the architecture *4-3-1*. This framework has four inputs, selected using the 2nd heuristic process, three hidden neurons in the hidden layer and one output variable. This model was chosen as the best model because it produced the best forecasts that in turn minimized the RMSE, MAE and MPE values during training compared to all the other simple ANN models. From Table 1, the architecture with the lowest value of RMSE, MAE and MPE are given in bold font.

Table 1: ANN Model Selection Summary

Input	Hidden Nodes	RMSE	MAE	MPE
1ˢᵗ Rule <1,2,3>	$h=1$	0.1838	2.37E-02	4.7350
	$h=2$	0.1738	2.06E-02	4.1291
	$h=3$	0.1736	1.97E-02	3.9313
	$h=4$	0.1624	3.35E-02	3.3510
	$h=5$	0.1657	3.27E-02	3.3438
	$h=6$	0.1657	3.44E-02	3.3438
	$h=7$	0.1600	3.16E-02	3.1602
2ⁿᵈ Rule<1,2,3,4>	$h=1$	0.0058	1.91E-05	0.0038
	$h=2$	0.0056	1.76E-05	0.0035
	$h=3$	**0.0038**	**8.07E-06**	**0.0016**
	$h=4$	0.0049	1.31E-05	0.0026
	$h=5$	0.0047	1.40E-05	0.0028
	$h=6$	0.0063	2.25E-05	0.0045
	$h=7$	0.0048	1.43E-05	0.0028
	$h=8$	0.0050	8.34E-05	0.0041
	$h=9$	0.0048	1.24E-05	0.0025
3ʳᵈ Rule<1>	$h=1$	0.2818	1.13E-02	-0.2958
	$h=2$	0.0288	1.37E-02	-0.3064
	$h=3$	0.0284	1.27E-02	-0.3038
3ʳᵈ Rule<1,2>	$h=1$	0.1851	4.83E-02	4.8338
	$h=2$	0.1859	4.81E-02	4.8083
	$h=3$	0.1880	4.65E-02	4.6491
	$h=4$	0.1761	4.04E-02	4.0382
	$h=5$	0.1667	4.04E-02	3.5793

3.2 EMD-ANN

Initially, the study found the original data to be broken down into six IMFs and residues. Next, as explained previously, each IMF underwent the ANN methodology. Starting by constructing the architecture based on the Kolmogorov's theorem for hidden nodes and Heuristc Method for input selection, the inputs are then normalized, partitioned and trained for learning before being forecasted for the model evaluation process. The weights are updated and adjusted by learning. The fitted values are summarized later for each data set to obtain the final fitted result.

Then, the accuracy of the hybrid approach is evaluated by measuring the error measure between the actual data and the result summarized as illustrated in Table 2. The three error measure is obtained based on the criterion pre-selected for the study: 1) RMSE = 0.00299, RMSE of zero is almost never achieved and a lower value nearing zero is considered the best performance.Next, 2) MAE = 0.00001272, the small value of MAE which is nearing zero indicates that the vertical distance between the actual data point and model line is of a small scale. This can further be explained by the third eror measure criterion, 3) MPE = 0.001024

The positive value of MPE indicates that the model consistently produces values that are lower than the actual data. Overall, it can be said that the model established is a good fit with minimal under-fitting.

Table 2: EMD-ANN Model Accuracy

	RMSE	MA	MP
EMD-AN	0.0029	1.27E−05	0.00102

3.3 Simple ANN and EMD-ANN Model Comparison

Here, the best model from the simple ANN and the EMD-ANN model was compared based on their error measures. From Table 3, the hybrid method using 20% testing data yields lower RMSE, MAE and also MPE compared to the simple model. The RMSE value shows a decrease of 0.012, the MAE value shows a decrease by 0.0002 and the MPE value by 0.0448 when ANN is introduced. In conclusion, the results illustrate a result that is similar to the early hypothesis, that is the fact that introducing EMD into a simple ANN model would significantly improve the method in forecasting monthly paddy prices.

Table 3: Model Performance Comparison

	RMSE	MA	MP
AN	0.0031	1.47E-05	0.001472
EMD-AN	**0.0029**	**1.27E-05**	**0.001024**

4 Conclusion and Recommendation

The aim of this study is to demonstrate the use of EMD in improving the ANN modeling for forecasting paddy price. Using the heuristic input selection method and Kolmogrov's theorem for tuning the number of neurons in the hidden layer, four basic architectures for the simple ANN model were defined. All the required computation was done, and as a result, the best simple ANN model was defined as the model with the *4-3-1* architecture. Next, EMD was introduced in the formulation of the model, whereby, the original data was decomposed into six IMFs. Next, a simple ANN methodology was conducted on each IMF. The fitted values obtained from the simple ANN of each IMF are then summed up to produce the final forecasts value. Based on these forecast values, the error measure of the EMD-ANN was obtained and compared with the simple ANN. The study shows that the EMD-ANN victoriously improves the forecasts of the paddy price proven by the minimized errors compared to the erorrs computed from the simple ANN.

Further research could be done by exploring into the various methods available in selecting the architecture such as the exhaustive search, forward selection, backward selection and many

more. It would also be of interest to explore into the possibility of using only 80% of the overall data for EMD [9] instead of 100% as has been carried out in this study.

Acknowledgments

References

[1] Abadan, S., & Shabri, A. Hybrid empirical mode decomposition-ARIMA for forecasting price of rice. *Applied Mathematical Sciences.* 2014. 8(63): 3133-3143.

[2] Rahim, F. H. A., Abidin, N. Z., & Hawari, N. N. (2017, November). Analyzing the impact of price subsidy on rice self-sufficiency level in Malaysia: A preliminary finding. In *AIP Conference Proceedings (Vol. 1905, No. 1*, p. 040025). AIP Publishing.

[3] Kusumaningsih, A., Jamhari, J., & Darwanto, D. H. Analysis of rice price trend and vertical integration of rice market in Indonesia. *Ilmu Pertanian (Agricultural Science).* 2016. 1(2): 074-079.

[4] Alexander, C. *Market Models: A Guide to Financial Data Analysis.* John Wiley & Sons. 2001.

[5] Aymanns, C., Caccioli, F., Farmer, J. D., & Tan, V. W. Taming the Basel leverage cycle. *Journal of Financial Stability.* 2016. 27: 263-277.

[6] Caccioli, F., Kondor, I., Marsili, M., & Still, S. Liquidity risk and instabilities in portfolio optimization. *International Journal of Theoretical and Applied Finance.* 2016. 19(05). 1650035.

[7] Clements, M. P., Franses, P. H., & Swanson, N. R. Forecasting economic and financial time-series with non-linear models. *International Journal of Forecasting.* 2004. 20(2): 169-183.

[8] Kim, K. J. Financial time series forecasting using support vector machines. *Neurocomputing.* 2003. 55(1-2): 307-319.

[9] Nava, N., Di Matteo, T., & Aste, T. Financial time series forecasting using empirical mode decomposition and support vector regression. *Risks.* 2018. 6(1): 7.

[10] Varga-Haszonits, I., Caccioli, F., & Kondor, I. (2016). Replica approach to mean-variance portfolio optimization. *Journal of Statistical Mechanics: Theory and Experiment.* 2016. (12). 123404.

[11] Zhang, G., Patuwo, B. E., & Hu, M. Y. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting.* 1998. 14(1): 35-62.

[12] Varga-Haszonits, I., Caccioli, F., & Kondor, I. Replica approach to mean-variance portfolio optimization. *Journal of Statistical Mechanics: Theory and Experiment.* 2016. 12. 123404.

[13] Crone, S. F., &Kourentzes, N. Input variable selection for time series prediction with neural networks-an evaluation of visual, autocorrelation and spectral analysis for varying seasonality. In *European Symposium on Time Series Prediction*. 2007. 195-205.

[14] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., ... & Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*. 454(1971): 903-995.

[15] Zhou, F., Zhou, H. M., Yang, Z., & Yang, L. EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*. 2019. 115: 136-151.

[16] Yu, L., Wang, S., & Lai, K. K. Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics*. 2008. 30(5): 2623-2635.

[17] Lin, C. S., Chiu, S. H., & Lin, T. Y. Empirical mode decomposition–based least squares support vector regression for foreign exchange rate forecasting. *Economic Modelling.* 2012. 29(6): 2583-2590.

[18] Cheng, C. H., & Wei, L. Y. A novel time-series model based on empirical mode decomposition for forecasting TAIEX. *Economic Modelling.* 2014. 36: 136-141.

[19] Lu, C. J., & Shao, Y. E. Forecasting computer products sales by integrating ensemble empirical mode decomposition and extreme learning machine. *Mathematical Problems in Engineering.* 2012.

[20] Chen, C. F., Lai, M. C., & Yeh, C. C. Forecasting tourism demand based on empirical mode decomposition and neural network. *Knowledge-Based Systems*. 2012. 26: 281-287.

[21] Chang, P. C. A novel model by evolving partially connected neural network for stock price trend forecasting. *Expert Systems with Applications*. 2012. 39(1): 611-620.

[22] Lu, C. J. Integrating independent component analysis-based denoising scheme with neural network for stock price prediction. *Expert Systems with Applications*. 2010. 37(10): 7056-7064.

[23] Omidi, A., Nourani, E., & Jalili, M. Forecasting stock prices using financial data mining and Neural Network. In *2011 3rd International Conference on Computer Research and Development (2011, March).* (Vol. 3, pp. 242-246). IEEE.

[24] Wang, J. Z., Wang, J. J., Zhang, Z. G., & Guo, S. P. Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*. 2011. 38(11): 14346-14355.

[25] Shabri, A., Samsudin, R., & Ismail, Z. Forecasting of the rice yields time series forecasting using artificial neural network and statistical model. *Journal of Applied Sciences.* 2009. 9(23): 4168-4173.

[26] Huang, W., Wang, S., Yu, L., Bao, Y., & Wang, L. A new computational method of input selection for stock market forecasting with neural networks. In *International Conference on Computational Science. (2006, May). (pp. 308-315).* Springer, Berlin, Heidelberg.

[27] Zhang, G. P., Patuwo, B. E., & Hu, M. Y. A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers & Operations Research.* 2001. 28(4): 381-396.

[28] Zhang, G., Patuwo, B. E., & Hu, M. Y. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting.* 1998. 14(1): 35-62.

[29] Walczak, S., &Cerpa, N. Heuristic principles for the design of artificial neural networks. *Information and Software Technology.* 1999. 41(2): 107-117.

[30] Rabie, H., El-Beltagy, M., Tharwat, A., & Hassan, S. Exploring Input Selection for Time Series Forecasting. In *DMIN.* 2008. 228-232.

[31] Sheela, K. G., & Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering.* 2013.

[32] Kusiak, A. *Computational Intelligence in Design and Manufacturing.* John Wiley & Sons. 2000.

[33] Hecht-Nielsen, R. Kolmogorov's mapping neural network existence theorem. In *Proceedings of the International Conference on Neural Networks (Vol. 3, pp. 11-14). (1987, June).* IEEE Press. New York.